



EMS HTTPS REST API

REST API version: 1.0.0

**HTTPS REST API
PROTOCOL**

Public version

Version 1.0.0

May 13, 2026

EMS Webapi (1.0.0)

Download OpenAPI specification: [Download](#)

REST API for the EMS smart-meter platform: read system data, browse the configuration (Devices → Loads → Signals) and get realtime / historical measurements.

Domain model

- **Device**: a physical or logical meter known to the gateway (e.g. the built-in main meter or a Modbus-connected EM24).
- **Load**: a measurement context that groups the signals produced by a single physical channel and carries protocol-specific configuration (`meterBus`, `tcp`, `rtu`, `virtual`, `P1`).
- **Signal**: a single quantity exposed by a Load (voltage, current, energy counter, ...) identified by a stable signal-identifier code.
- **Realtime**: the latest sampled value of a Signal.
- **History**: the historical sampled value of a Signal.

Authentication

Every endpoint except `GET /ver` and `POST /v1/auth/login` requires a Bearer JWT obtained from the login endpoint. Tokens expire 30 minutes after issue.

Pass the token in the `Authorization` header:

```
Authorization: Bearer <token>
```

Login

Create a POST request with JSON body with username and password of a API role account. The returned token expires 30 minutes after issue and must be sent as **Authorization: Bearer <token>** on every protected endpoint.

REQUEST BODY SCHEMA: application/json
required

username required	string
password required	string <password>

Responses

> 200 Login successful

> 401 Missing or invalid token

POST /v1/auth/login

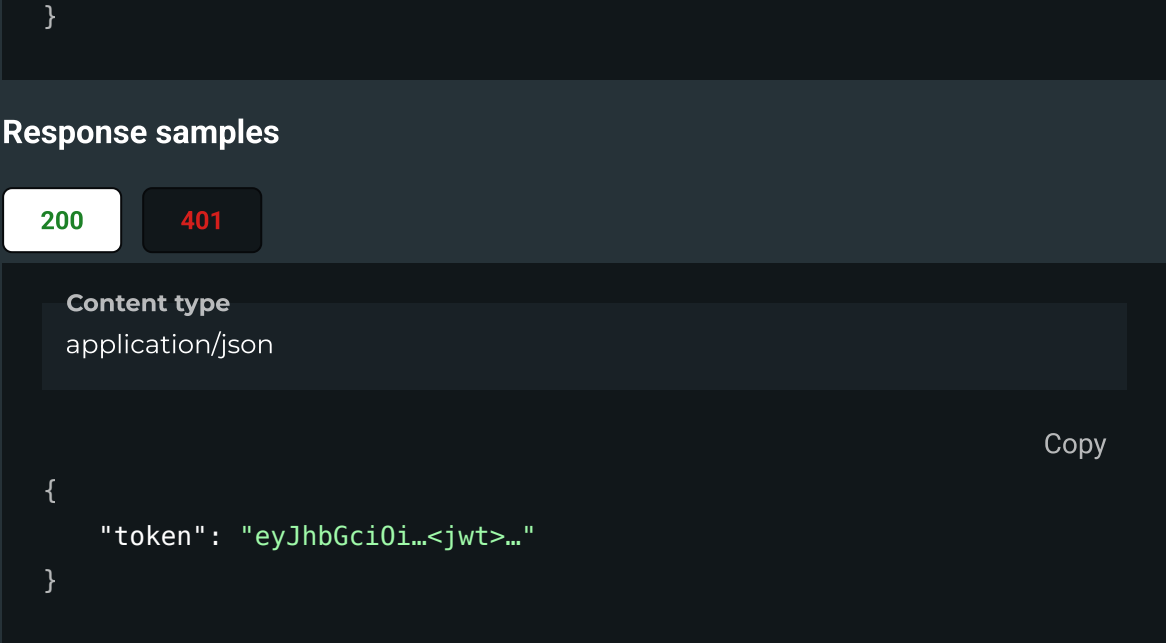
Request samples

Payload

Content type
application/json

Copy

```
{  
  "username": "admin",  
  "password": "s3cret!"  
}
```



Get API version

Returns the version of this API. Public – no token required.

Responses

➤ **200** Current API version



Copy

```
{  
  "api_ver": 1  
}
```

Get EMS serial number

AUTHORIZATIONS: > *bearerAuth*

Responses

> 200 Serial number

> 401 Missing or invalid token

GET /v1/serial-number



Response samples

200

401

Content type

application/json

Copy

```
{  
  "sn": "BX10000000"  
}
```

Get the EMS MAC address

AUTHORIZATIONS: > *bearerAuth*

Responses

> 200 MAC address

> 401 Missing or invalid token

The screenshot shows a REST client interface with a dark theme. At the top, a method dropdown is set to 'GET' and the URL is '/v1/mac'. Below this, the 'Response samples' section has two tabs: '200' (selected) and '401'. The '200' tab displays the response details: 'Content type' is 'application/json' and the response body is a JSON object:

```
{  "mac": "00:1A:2B:3C:4D:5E"}
```

. A 'Copy' button is located to the right of the JSON body.

Get EMS Firmware version

AUTHORIZATIONS: > *bearerAuth*

Responses

> 200 Firmware version

> 401 Missing or invalid token

GET /v1/firmware

Response samples

200

401

Content type

application/json

Copy

```
{
  "fw": "EMS_1.0.0.0.250101.000000"
}
```

Get EMS Part number

AUTHORIZATIONS: > *bearerAuth*

Responses

> 200 Part number

> 401 Missing or invalid token



GET /v1/part-number

Response samples

200 401

Content type
application/json

Copy

```
{
  "pn": "EMS10AV5S1X"
}
```

Get current local date and timezone

AUTHORIZATIONS: > *bearerAuth*

Responses

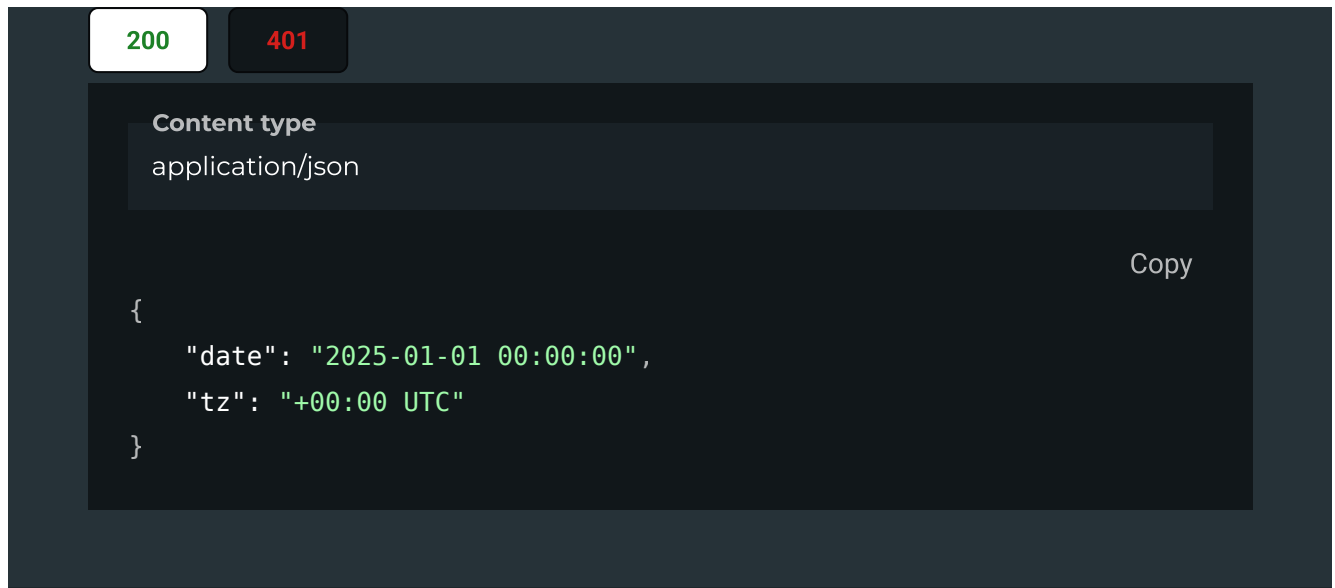
> 200 Date and timezone

> 401 Missing or invalid token



GET /v1/date

Response samples



List devices

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

uids	string Example: <code>uids=1, 2, 59</code> Comma-separated list of UIDs to filter by.
protocol	string Enum: <code>"rtu"</code> <code>"tcp"</code> <code>"meterBus"</code> <code>"virtual"</code> <code>"P1"</code> Filter by protocol. Accepted values: <code>rtu</code> , <code>tcp</code> , <code>meterBus</code> , <code>virtual</code> , <code>P1</code> .

Responses

> **200** Device map keyed by UID

> **400** Malformed request (e.g. invalid query parameter)

> **401** Missing or invalid token

GET

/v1/devices

▼

Response samples

200

400

401

Content type

application/json

Copy

Expand all

Collapse all

```
{
  - "devices": {
    + "1": { ... },
    + "59": { ... }
  }
}
```

Get a device by UID

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
required integer
Example: 1
Resource UID

Responses

> 200 Device

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/devices/{uid}

Response samples

200

401

404

Content type
application/json

Copy

```
{
  "uid": 1,
  "label": "Main meter",
  "partNumber": "EMS10AV5S1X",
  "system": "1P"
}
```

List loads

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

uids

string

Example: `uids=1, 2, 59`

Comma-separated list of UIDs to filter by.

protocol

string

Enum: `"rtu"` `"tcp"` `"meterBus"` `"virtual"` `"P1"`

Filter by protocol. Accepted values: `rtu`, `tcp`, `meterBus`, `virtual`, `P1`.

Responses

> **200** Load map keyed by UID

> **400** Malformed request (e.g. invalid query parameter)

> **401** Missing or invalid token

GET /v1/loads

Response samples

200

400

401

Content type

application/json

Example

Built-in main meter (Meter-Bus)

Copy Expand all Collapse all

```
{
  - "loads": {
    + "2": { ... }
  }
}
```

Get a load by UID

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
required integer
Example: 1
Resource UID

Responses

> 200 Load

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/loads/{uid}

Response samples

200

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  "uid": 2,
  "label": "Main",
  "enable": true,
  "index": 1,
  "type": 0,
  "did": 1,
  "meterBus": { },
  "tcp": { },
  "rtu": { },
  "virtual": { },
  "P1": { }
```

```
}
```

List loads belonging to a device

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

uid
required integer
Example: 1
Resource UID

Responses

> 200 Load map keyed by UID

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/devices/{uid}/loads

Response samples

200

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  - "loads": {
    + "2": { ... }
  }
}
```

List signals

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

uids	string Example: <code>uids=1, 2, 59</code> Comma-separated list of UIDs to filter by.
limit	integer Default: <code>1000</code> Example: <code>limit=100</code> Maximum number of records to return. Default 1000.
page	integer Default: <code>1</code> Example: <code>page=1</code> 1-based page number. Default 1.

Responses

> **200** Signal map keyed by UID

> **400** Malformed request (e.g. invalid query parameter)

> **401** Missing or invalid token

GET

/v1/signals

▼

Response samples

200

400

401

Content type

application/json

Copy

Expand all

Collapse all

```
{
  - "signals": {
    + "100": { ... },
    + "101": { ... }
  }
}
```

Get a signal by UID

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
required integer
Example: 1
Resource UID

Responses

> 200 Signal

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/signals/{uid}

Response samples

200

401

404

Content type

application/json

Copy

```
{
  "uid": 100,
  "si": 29,
  "sir": 29,
  "enable": true,
  "lid": 83,
  "did": 82
}
```

List signals belonging to a device

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
required integer
Example: 1
Resource UID

Responses

> 200 Signal map keyed by UID

> 401 Missing or invalid token

> 404 Resource not found

GET

/v1/devices/{uid}/signals

▼

Response samples

200

401

404

Content type

application/json

Copy

Expand all

Collapse all

```

{
  - "signals": {
    + "100": { ... },
    + "101": { ... }
  }
}

```

List signals belonging to a load

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid integer
required Example: 1

Resource UID

Responses

> **200** Signal map keyed by UID

> **401** Missing or invalid token

> **404** Resource not found

GET /v1/loads/{uid}/signals

Response samples

200

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  - "signals": {
    + "100": { ... },
    + "101": { ... }
  }
}
```

Latest realtime values for many signals

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

uids	string Example: <code>uids=1, 2, 59</code> Comma-separated list of UIDs to filter by.
limit	integer Default: <code>1000</code> Example: <code>limit=100</code> Maximum number of records to return. Default 1000.
page	integer Default: <code>1</code> Example: <code>page=1</code> 1-based page number. Default 1.

Responses

> **200** Realtime map keyed by signal UID

> **400** Malformed request (e.g. invalid query parameter)

> **401** Missing or invalid token

GET

/v1/signals/realtime

▼

Response samples

200

400

401

Content type

application/json

Copy

Expand all

Collapse all

```
{
  - "realtime": {
    + "100": { ... },
    + "101": { ... }
  }
}
```

Latest realtime value for a single signal

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
 required integer
 Example: 1
 Resource UID

Responses

> 200 Realtime sample

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/signals/{uid}/realtime

Response samples

200

401

404

Content type

application/json

Copy

```
{
  "uid": 100,
  "si": 29,
  "v": 230.4,
  "t": 1752493029975,
  "s": 1,
  "r": 0
}
```

Latest realtime values for all signals of a device

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

uid
required

integer
Example:
Resource UID

Responses

> 200 Realtime map keyed by signal UID

> 401 Missing or invalid token

> 404 Resource not found

GET

/v1/devices/{uid}/signals/realtime

▼

Response samples

200

401

404

Content type

application/json

Copy

Expand all

Collapse all

```
{
  - "realtime": {
    + "100": { ... },
    + "101": { ... }
  }
}
```

Latest realtime values for all signals of a load

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
required integer
Example: 1
Resource UID

Responses

> 200 Realtime map keyed by signal UID

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/loads/{uid}/signals/realtime

Response samples

200

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  - "realtime": {
    + "100": { ... },
    + "101": { ... }
  }
}
```

Historical sample for many signals

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

uids	string Example: <code>uids=1, 2, 59</code> Comma-separated list of UUIDs to filter by.
start	string Example: <code>start=1735689600000</code> Start of the time range — Unix timestamp in milliseconds .

end	string Example: <code>end=1735776000000</code> End of the time range — Unix timestamp in milliseconds .
limit	integer Default: <code>1000</code> Example: <code>limit=100</code> Maximum number of records to return. Default 1000.
page	integer Default: <code>1</code> Example: <code>page=1</code> 1-based page number. Default 1.

Responses

> **200** History map keyed by signal UID

> **400** Malformed request (e.g. invalid query parameter)

> **401** Missing or invalid token

GET

/v1/signals/history

▼

Response samples

200

400

401

Content type

application/json

Copy Expand all Collapse all

```
{
  - "history": {
    + "3": [ ... ],
    + "4": [ ... ]
  }
}
```

Historical sample for a single signal

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
required integer
Example: 1
Resource UID

QUERY PARAMETERS

→ start string
Example: start=1735689600000
Start of the time range — Unix timestamp in **milliseconds**.

→ end string
Example: end=1735776000000
End of the time range — Unix timestamp in **milliseconds**.

→ limit integer
Default: 1000
Example: limit=100
Maximum number of records to return. Default 1000.

→ page integer
Default: 1
Example: page=1
1-based page number. Default 1.

Responses

> 200 History array

> 400 Malformed request (e.g. invalid query parameter)

> 401 Missing or invalid token

> 404 Resource not found

The screenshot shows a REST client interface with a dark theme. At the top, a dropdown menu displays 'GET /v1/signals/{uid}/history'. Below this, the 'Response samples' section has four buttons: '200' (highlighted in white), '400', '401', and '404'. Under the '200' button, the 'Content type' is set to 'application/json'. To the right of the JSON content are three buttons: 'Copy', 'Expand all', and 'Collapse all'. The JSON response is displayed in a dark box with syntax highlighting.

```
[
  - {
    "uid": 3,
    "t": 1735689659,
    "p": 1735689660,
    "v": 10.5,
    "d": 0,
    "max": 100,
    "min": 0
  }
]
```

Last historical sample for many signals

AUTHORIZATIONS: > *bearerAuth*

QUERY PARAMETERS

uids	string Example: <code>uids=1, 2, 59</code> Comma-separated list of UIDs to filter by.
limit	integer Default: <code>1000</code> Example: <code>limit=100</code> Maximum number of records to return. Default 1000.
page	integer Default: <code>1</code> Example: <code>page=1</code> 1-based page number. Default 1.

Responses

> **200** Map of last history samples keyed by signal UID

> **400** Malformed request (e.g. invalid query parameter)

GET /v1/signals/history/last

Response samples

200

400

Content type

application/json

Copy Expand all Collapse all

```
{
  - "history": {
    + "3": { ... },
    + "4": { ... }
  }
}
```

Last historical sample for a single signal

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

uid
required integer
Example: 1
Resource UID

Responses

> 200 Last history sample

GET /v1/signals/{uid}/history/last

Response samples

200

Content type
application/json

Copy

```
{
  "uid": 3,
  "t": 1735689659,
  "p": 1735689660,
  "v": 10.5,
  "d": 0,
  "max": 100,
  "min": 0
}
```

Historical sample for all signals of a device

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

uid
required integer
Example: 1
Resource UID

QUERY PARAMETERS

start string
Example: start=1735689600000
Start of the time range – Unix timestamp in **milliseconds**.

end string
Example: end=1735776000000
End of the time range – Unix timestamp in **milliseconds**.

limit integer
Default: 1000
Example: limit=100
Maximum number of records to return. Default 1000.

page

integer

Default:

Example:

1-based page number. Default 1.

Responses

> **200** History map keyed by signal UID

> **400** Malformed request (e.g. invalid query parameter)

> **401** Missing or invalid token

> **404** Resource not found

GET /v1/devices/{uid}/signals/history

Response samples

200

400

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  - "history": {
    + "3": [ ... ],
    + "4": [ ... ]
  }
}
```

Last historical sample for each signal of a device

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

uid
required integer
Example: 1
Resource UID

Responses

> 200 Map of last history samples keyed by signal UID

> 401 Missing or invalid token

> 404 Resource not found

GET /v1/devices/{uid}/signals/history/last

Response samples

200

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  - "history": {
    + "3": { ... },
    + "4": { ... }
  }
}
```



```
}
```

Historical sample for all signals of a load

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

uid
required
integer
Example: 1
Resource UID

QUERY PARAMETERS

start
string
Example: start=1735689600000
Start of the time range — Unix timestamp in **milliseconds**.

end
string
Example: end=1735776000000
End of the time range — Unix timestamp in **milliseconds**.

limit
integer
Default: 1000
Example: limit=100
Maximum number of records to return. Default 1000.

page
integer
Default: 1
Example: page=1
1-based page number. Default 1.

Responses

> 200 History map keyed by signal UID

> 400 Malformed request (e.g. invalid query parameter)

> 401 Missing or invalid token

> 404 Resource not found

GET
/v1/loads/{uid}/signals/history

Response samples

200
400
401
404

Content type

application/json

Copy Expand all Collapse all

```

{
  - "history": {
    + "3": [ ... ],
    + "4": [ ... ]
  }
}

```

Last historical sample for each signal of a load

AUTHORIZATIONS: > *bearerAuth*

PATH PARAMETERS

→ uid
 required integer
 Example: 1
 Resource UID

Responses

> **200** Map of last history samples keyed by signal UID

> **401** Missing or invalid token

> **404** Resource not found

GET /v1/loads/{uid}/signals/history/last

Response samples

200

401

404

Content type

application/json

Copy Expand all Collapse all

```
{
  - "history": {
    + "3": { ... },
    + "4": { ... }
  }
}
```