AN-123: Daily Report API Documentation

This document describes the API calls you can use to authenticate, access your Company Site List (CSL) or Personal Site List (PSL) and get readings data from each of your sites.

All API calls will be presented with cURL examples. If copying from this document, beware some line breaks may turn into spaces

Authentication

To use the API, you must already have a Tehama account. Our CIT software is used by CSL administrators to create user accounts, including API users. If you need an account, contact your site administrator. If you are a small user with only a few sites, you would have already created a PSL when you first used the CIT. In both cases you use your CIT login credentials for the API.

Our base URL is https://api-v3.tehamawireless.net

To authenticate, make a POST to the base URL + /auth/login. Within the data field you will add your credentials in the form {'username': 'user', 'password': 'pass'}.

Here is a cULR example:

```
curl -X POST -H 'Content-Type: application/json' -d '{"username": "user", "password": "pass"}' https://api-v3.tehamawireless.net/auth/login
```

The response is a 300+ character token

{"token":" eyJraWQiOiJObDQ5 aTZOWjVKYjM2QzdvZ1F1RDdIdmRWSU5iOFNTc1F6UVMxaU1GV
lVNPSIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiI3ZWY2OGE5ZC01YzIwLTRmZDkt

[[about 10 more lines like this …]]

gUfPvt-4BcVgWt2yQrlJ5alBQuVPS6tBigBr8AfiT4oprHx-8BjqgIs1sL5g-avtQg"}

Capture the token value for later use.

Get the Site List

To get your Site List, make a GET to the base URL + /tehama/site-list-server. The token goes in the header associated with X-Authorization.

Using cURL:

```
curl -X 'GET' -H 'Accept: application/json' -H 'X-Authorization: eyJraWQiOiJObD . . .avtQg' https://api-v3.tehamawireless.net/tehama/site-list-server
```

This returns all the entries in your Site List in JSON format. Here is an example of a single entry. You will get a block like this for each property in your Site List:

```json
{
    "access_code": "88471391331",
    "address": "",
    "city": "Pembroke",
    "dcap_ip_adrs": "",
    "get_port_num": "false",
    "knocker_dly": "5",
    "knocker_dly_enbld": "False",
    "name": "Elan Flower Mound #12294 (28667)",
    "nodeId": "614",
    "nodeType": "DCAP",
    "note": "",
    "parent_node_id": "4462",
    "property_id": "4706362",
    "sql_user_name": "cituser",
    "state": "NC",
    "zipcode": "28372"
},
```

The primary field to extract from this is the access_code. Use the access_code for each entry to feed the next API call to collect the metering data.

Most of the other fields are populated when you commission a property using our software or are generated by the DCAP itself. **In most cases, extracting a list of access_code fields is the primary use for making this call.** If you maintain a separate list of your properties, you may skip this call.

Get the Reads Data

For each property (that is access_code), a call is made to the report-data API call. This returns the data point(s) for each MDT associated with the property.

The API call requires at a minimum the access_code value, but you can also specify a specific deviceID (MDT) to limit the output to a single device, or a start_time and end_time if you wish to collect historical data. Both deviceID and start/end times can be specified together. If no deviceID or time stamps are specified, the most recent data point for each meter/MDT is returned.

The URL for the report data is the base URL + /tehama/sites/report-data/ + AC.

cURL example for latest data (no no device or date parameters)

AN-123: Daily Report API Documentation

curl -X 'GET' -H 'Accept: application/json' -H 'X-Authorization: eyJraWQiOiJObD
. . .avtQg' https://api-v3.tehamawireless.net/tehama/sites/report-data/88471391331

The output for a single MDT looks like this:

```
{
    "Addr": "",
    "Apt": "68 ",
    "BattVolt": "3020",                         #Measured in millivolts
    "Building": "1",
    "FarEndRssi": "77",                         # Receive Signal Strength of this unit as heard by the Link Partner,
                                                range is 0-100
    "LP": "e3000a78",                           # LP is Link Partner, the Repeater or DCAP this device is
                                                communicating with.
    "LQ": "98",                                 # Range is 0-100, 100 is best
    "LatestLinkStatusDataReadTime": "2022-03-02 04:51:12",
    "LocationNote": "Vacant",
    "NodeId": "5532",
    "PoweredDownUnits": "false",                # Is this unit in a Powered Down state
    "RSSI": "88",                               # Receive Signal Strength of the Link Partner's signal, range is
                                                0-100
    "RadioID-Dec": "2175922659",
    "RadioID-Hex": "81B1F1E3",
    "readings": [
     {
        "RawReading": 47190,                    # The raw MDT count or reading
        "ReadTime": "2022-03-02 04:29:41",
        "Reading": 491900,                      # Adjusted reading using the formula Reading = initial_meter_read
                                                + (RawReading * factor)
        "alerts": [
         {
            "alert_type": "Daily_Alert",
            "enabled": "true",
            "recipient_list": "daily_email_adrs"
         }
        ],
```

```
            "display_type": "Encoded_Meter_1",
            "factor": "10.000",                    # Count factor multiplication of RawReading value
            "initial_meter_read": "20000",         # The initial meter reading for a pulse meter. MDT count should be
                                                     zero at time of reading

            "initial_meter_time": "",
          "meter_serial_number": "0000203389",
            "meter_style": "A",   # A = All, C = cold, H =hot, O = Other
            "meter_type": "W",  # W = Water, G = Gas, E = Electric, B = BTU
            "mon_pt": "0",
            "note": "3/4 Neptune T-10",
            "sensor_type": "53",                   # See below
            "units": "Gallons"
        }
      ]
    },
```

cURL example with Start and End times

The date format follows the ISO 8601 format, using the "basic" time format: YYYY-MM-DDThhmm. ALL TIMES are UTC.

```
curl -X 'GET' -H 'Accept: application/json' -H 'X-Authorization: eyJraWQiOiJObD . . .avtQg' 'https://api-v3.
    tehamawireless.net/tehama/sites/report-data/88471391331?start_time=2022-02-01T1000&end_
    time=2022-02-01T1130'
```

Note that the URL needs to be in quotes when using the start/end times when using cURL or the & needs to be escaped (\&).

cURL example with deviceID

A deviceID is the 8 digit HEX character serial number of a Tehama radio. The deviceID start with "8" for any Tehama MDT or with "A" for an Inovonics PMT.

```
curl -X 'GET' -H 'Accept: application/json' -H 'X-Authorization: eyJraWQiOiJObD . . .avtQg' https://api-v3.
    tehamawireless.net/tehama/sites/report-data/88471391331?device_id=83D01234
```

The start_time, end_time, and device_id URL modifiers can be in any order.

Each Data Range call is limited to 24 hours (except when you specify a specific RadioID). If you need a week of data, you will need to make seven calls, one for each day.

When date ranges are used, you will receive ALL the data between the two times specified. Therefore, for a typical 24 hour call, you will 24 separate readings [] arrays for each meter. If you want to limit this, narrow the time window to one or two hours.

Timestamps returned by the API are in UTC time zone. At the very end of the JSON data is an array of site_details that includes the timezone field that can be used to adjust the timestamps to local time of the property.

```
"site_details": {
    "city": "Washington",
    "property_id": "34as543",
    "site_name": "Union Square",
    "state": "DC",
    "timezone": "EST5EDT4",
    "zipcode": "20002"
}
```

sensor_type definitions

The following table describes the various "sensor" types coming from a Tehama system. For most sites with single meter MDTs, you can ignore this. It does become critical for Dual Input MDT (hot & cold usually) or where Temperature or other sensors are used. For a dual Pulse MDT for example, you will receive a pair of readings [] arrays and you will use the sensor_type field to distinguish the two.

Sensor Type	Decimal	Hex
Internal Temperature	1	0x01
External Temperature	2	0x02
Contact Closure (0 = closed)	8	0x08
Sanding Leak status (1 = Alarmed)	48	0x30
Test Totalizer (used by development only)	80	0x50
Pulse_Totalizer_1	81	0x51
Pulse_Totalizer_2	82	0x52
Encoded_Meter_1	83	0x53
Hersey_Totalizer	84	0x54
ToH Totalizer	85	0x55
RunTime_Counter (24VAC)	88	0x58
RunTime_lineCut (24VAC) (1 = wires are not connected properly to certain Zone Valves)	89	0x59
RunTime_P1 (Relay)	91	0x5B
Runtime_P2 (Relay)	92	0x5C
Encoded_Meter_2	93	0x5D
GWF_Allread Encoder_1	113	0x71
GWF_AllRead Encoder_2	114	0x72
PMT_Sensor	253	0xFD