Also Energy
Production Data API

Version 1.8.0
5/16/2016

Also Energy, Inc
3360 Mitchell Lane, Suite A
Boulder, CO 80301
Phone (866) 303-5668

Table of Contents

1

Overview

The AlsoEnergy API allows developers to integrate their own applications, websites and back-office systems with the AlsoEnergy platform. This interface provides a convenient means to programmatically access data from all your sites and devices.

The Production Data API is implemented using standard web service technologies and is compatible with all common development environments. Sample source code and technical support is available to assist with development.

An additional Admin API is available for access to sites, hardware and other information pertaining to the configuration of PowerTrack accounts. Contact AlsoEnergy for more information.

2

Getting Started

The web service description is available at:

http://www.alsoenergy.com/WebAPI/WebAPI.svc?wsdl

The web service consists of functions and classes that encapsulate data for communication to and from the server. The basic operations are:

Login – Establishes a session with the server. All subsequent operations require the session key that is returned by the Login function.

GetSiteList – Returns list of site IDs and names.

GetSiteHardwareList – Returns list of hardware IDs and names.

GetBinData – Returns device data.

All functions return an object based on a **Result** object, containing a status code. The following list shows all possible return codes:

```
OK = 200
AuthenticationFailed = 4000
SecureConnectionRequired = 4001
LoginFailed = 4002
NoAPIAccess = 4003
NoData = 4004
AccessDenied = 4005
ServerTemporaryUnavailable = 5000,
ServerError = 5001
InvalidInputData = 5002
UsageLimitExceeded = 5003
HardwareNotConfigured = 5004
```

NOTE: This API uses SOAP (Simple Object Access Protocol). All parameters listed below are case sensitive. All date/time parameters are to be entered in local site time unless otherwise specified. Date/time strings are in the following formats:

MM/dd/yyyy (i.e. 12/25/2012)

yyyy-MM-ddTHH:mm:ss (2015-12-01T14:00:30)

Restrictions

User accounts require special permission to access this API. Please contact Also Energy for a username and password.

In order to prevent excessive server load, access to the web service is limited to 60 calls per hour.

The data upload capabilities of the API are restricted by account permission. Please contact Also Energy to enable data upload through the Production Data API.

Additional functions are also available for accessing administrative features of the system, such as customer, site and hardware configurations. Please contact Also Energy if administrative data access is required.

3

Functions

Login

The Login establishes a session with the server. The resulting session is identified by the returned SessionID. All other API functions require a valid session ID. The session expires if no calls are made for more than 20 minutes, at which point the session ID is invalid and another call to Login is required to acquire a new session ID.

For security reasons, the Login call must be made over HTTPS. Either HTTP or HTTPS may be used for all other calls (HTTPS results in a 30% increase in the amount of data transferred, so HTTP is recommended when requesting large amounts of data).

Parameters

username (string)
password (string)

Returns

LoginResult (class)
    Code (int)
    SessionID (string)

GetSiteList

The GetSiteList function returns a list of site IDs and Names.

Parameters

sessionID (string)

Returns

ListResult (class)
- Code (int)
- ListItem[] (array of ListItem objects):
  - Site object:
    - ID (int)
    - Type (int)
    - Name (string)

GetSiteHardwareList

The GetSiteHardwareList function returns a list of information about all devices located on a site. This information is needed in order to query for data.

Parameters

sessionID (string)
siteID (int)

Returns

HardwareResult (class)
- Code (int)
- HID (int) – Hardware ID
- FunctionCode (string) – Identifies the device type:
  - PV – Inverter
  - CM – Consumption meter
  - GM – Grid meter
  - PM – Production meter
  - SA – Sub array current sensors
  - SC – String combiner
  - SS – Solstice string combiner
  - TS – Thermal storage
  - TP – Thermal production
  - WS – Weather station
- Description (string) – Display name
- ArchiveFields [] (array of FieldInfo objects):
  - Name (string) – field name
  - Units (string) – Data units
- SerialNum(string) – Serial number of the hardware

GetSummaryHardware

The GetSummaryHardware function returns information about a virtual summary device that exists for each site within PowerTrack. This device references daily summary data about the site such as total kWh produced and insolation. It provides easier and faster access when querying for total site production using one day or larger bins. Use the GetSummaryData function to query for data from the summary hardware.

Parameters

sessionID (string)
siteID (int)

Returns

HardwareResult (class)
- Code (int)
- HID (int) – Hardware ID.
- FunctionCode (string) – Identifies the device type:
  - SU – Summary
- Description (string) – Display name
- ArchiveFields [] (array of FieldInfo objects):
  - Name (string) – field name
  - Units (string) – Data units
- SerialNum(string) – Serial number of the hardware

GetBinData

The GetBinData function returns sample data from a list of devices and fields. Bins can range in size from 15 minute through 1 year.

Parameters

sessionID (string)
fromLocal (string) – date/time for the beginning of the requested data set.
NOTE: date/time strings are in US format (M/D/YYYY, i.e. 12/25/2012)
toLocal (string) – date/time for the end of the requested data set.
binSize (int) – The bin size is represented by the typical number of minutes per data sample, although the actual bin may be calculated from a variable number of samples as appropriate for the time period.
BinRaw (0) – Raw sample data
Bin15min (15) – 15 minutes
Bin1Hour (60) – One hour
BinDay (1440) – One day (23, 24 or 25 hours)
BinMonth (44640) – One month (28, 30 or 31 days)
BinYear (527040) – One year (365 or 366 days)
Fields [] (array of DataField objects):
HID (int) – Hardware ID
FieldName (string)
Function – Operation to apply to the data when calculating bins:
Avg (0) – Average of data points
Last (1) – Last sample for the period
Min (2) – Minimum sample for the period
Max (3) – Maximum sample for the period
Diff (4) – Difference between the last value for the period and the last value for the previous period.
Sum (5) – Sum of all samples for the period
Integral (6) – Integrate samples (i.e. calculates kWh from kW)

Returns

DataResult (class)
Code (int)
DataSet (array of DataBin objects):
Timestamp – string
Data [] (array of float) – one value per requested DataField.

GetSummaryData

The GetSummaryData function returns daily site summary data from a list of summary devices and fields. Bins can range in size from one day through 1 year.

Parameters

sessionID (string)
fromLocal (string) – date/time for the beginning of the requested data set.
toLocal (string) – date/time for the end of the requested data set.
binSize (int) – The bin size is represented by the typical number of minutes per data sample, although the actual bin may be calculated from a variable number of samples as appropriate for the time period.

- BinDay (1440) – One day (23, 24 or 25 hours)
- BinMonth (44640) – One month (28, 30 or 31 days)
- BinYear (527040) – One year (365 or 366 days)

Fields [] (array of DataField objects):

- HID (int) – Summary Hardware ID
- FieldName (string)
- Function – Operation to apply to the data when calculating bins:
  - Avg (0) – Average of data points
  - Last (1) – Last sample for the period
  - Min (2) – Minimum sample for the period
  - Max (3) – Maximum sample for the period
  - Diff (4) – Difference between the last value for the period and the last value for the previous period.
  - Sum (5) – Sum of all samples for the period
  - Integral (6) – Integrate samples (i.e. calculates kWh from kW)

Returns

DataResult (class)

- Code (int)
- DataSet (array of DataBin objects):
  - Timestamp – string
  - Data [] (array of float) – one value per requested DataField.

GetAlerts

The GetAlerts function returns alerts for a specified time period via the start and end dates. Users can specify which sites to get alerts for, otherwise alerts for all sites under the customer will be returned.

Parameters

sessionID (string)
start (string) – date/time for the start of the requested period for alerts
end (string) – date/time for the end of the requested period for alerts
siteID (int)

Returns

AlertResult (class)
- SiteID (int)
- AlertID (int)
- SubCode (int) – identifies the type of alert
- State (AlertStates) –
  - Undefined (0)
    - Detected (1) – alert condition detected but may not be verified, no action taken (no email)
    - Cancelled (2) – detected condition no longer exists (no email)
    - Pending (3) – alert condition is valid (email sent)
    - Resolved (4) – alert condition has been resolved
    - Ending (5) – alert condition is no longer valid but may not be verified
- HID (int) – Hardware ID
- Title (string) – Error message for the alert
- Timestamp (DateTime) – the date/time the alert triggered
- TimestampResolved (DateTime) – the date/time the alert was resolved. If the alert has not been resolved, this will show the current date and time.
- Message (string) – a detailed error message for the alert

4

Functions – Data Upload

The Data Upload API can only be used with accounts that do not have Guest permission. Contact Also Energy to change your API account credentials if you want to upload data.

Uploading data to a device involves two steps, first information about a gateway and the associated hardware is acquired, and then data is uploaded.

GetGatewayConfig

The GetGatewayConfig function returns a list of details for all devices associated with the requested gateway or data logger.

Parameters

sessionID (string) – A valid session ID is required (from the Login function).
gatewayID (string) – The GatewayID is assigned when the gateway device is created on the website. This is usually the MAC address for the gateway, although any unique identifier may be used.

Returns

GatewayConfigResult (class)
ServerTime (DateTime) – This is used to synchronize the gateway time with the server.
Devices (array of DeviceConfig objects):
DeviceType (string) – "Gateway", "Modbus", etc
HID (string) – Hardware ID
Address (int) – Device address (generally a Modbus or TCP/IP address).
Port (int) – Port number, for devices with multiple I/O ports.
Baud (int) – Baud rate to use to communicate with the device.
ComType (enum) – Physical connection type. One of:

Undefined,
RS232,
RS485_2Wire,
RS485_4Wire,
TCP

SerialNumber (string) – Gateway serial number
Description (string)
Params (list of NameVal objects) – Device-specific parameters.
Registers (list of RegInfo objects) – Information about each device register:

RegNum – Register number
RegName – Register name
DataType – Register data type:

Int,
Float,
String

NumBytes – Number of bytes in the register.

ArraySN (array of string) – Serial numbers for all devices in the array (for arrays of devices only).
PollDelaySeconds – Number of seconds between the data capture cycle for all devices.

UploadRawData

The UploadRawData function is used to upload one or more sets of data for one or more devices.

Parameters

sessionID (string) – A valid session ID is required (from the Login function).
data (List of UploadData objects):

HID (string) – Hardware ID (string or integer)

Timestamp (DateTime) – Timestamp for the data. Must be in UTC using the format "yyyy-MM-ddTHH:mm:ssZ" (2015-12-01T14:00:30Z)

IsLog (bool) – true if this data is from an earlier time, false for current data

NotResponding (bool) – true if the device did not reply when data was requested.

Data (string) – Comma-separated list if data for all registers.

Returns

UploadResult (class) –

UploadCount – Number of rows of data uploaded.

GetTimestamp

The GetTimestamp function returns time information for a device.

Parameters

sessionID (string) – A valid session ID is required (from the Login function).
hardwareID (string) – A device hardware ID.

Returns

TimestampResult (class) –

ServerUTC – The current UTC time on the server (should be within a few seconds of the current UTC time).

LastData – The timestamp for the most recently uploaded data.

LastUpdate – The timestamp for when the most recent upload the device took place.

SendHeartbeat

Provides current information about a gateway to the server and returns commands and a collection of firmware that the gateway should download and install.s

Parameters

sessionID (string) – A valid session ID is required (from the Login function).
gatewayID (string) – A device gateway ID (usually the MAC address).
data (Heartbeat (class)) – Heartbeat information:

```
class Heartbeat
{
	public string GWtype;		// Gateway type
	public string Model;		// Device OS details
	public string Version;
	public string Build;

	public string [] FirmwareVer;
	public string IPaddress;
	public DateTime LastBoot;	// UTC
	public NameVal [] Params;	// Device-specific parameters
}
```

Returns

HeartbeatResult (class) -

```
class HeartbeatResult : Result
{
	public string [] DownloadFirmware;
	public string Command;
	public string CommandParam;
	public string CommandID;
}
```

5

Standard Fields

Each type of device has a set of standard fields. Depending on the specific device, additional fields may also exist or some of the standard fields may be omitted. The GetSiteHardwareList function will return all available fields for a specific piece of hardware.

Inverters (PV)

KwAC – Power output

KwDC – Power input

KwhAC – Energy output

KwhACraw – Energy output (uncorrected)

Vdc – DC voltage input

Idc – DC current input

Vac – Average AC output voltage (line to neutral)

Iac – Average AC phase current.

Power Meters (PM, CM, GM, XM, WT)

KW – Power output

varTotal – VAR's

vaTotal – VA's

PowerFactor

PFA – Power factor, phase A

PFB – Power factor, phase B

PFC – Power factor, phase C

freq – Line frequency

KWHrec – Energy received

KWHdel – Energy delivered

KWHnet – Net energy

KWHloss – Energy loss (calculated)

KWHtot – Total energy

VacA – Phase A voltage

VacB – Phase B voltage

VacC – Phase C voltage

VacAB – Phase voltage, A to B

VacBC – Phase voltage, B to C

VacCA – Phase voltage, C to A

IacA – Phase A current

IacB – Phase B current

IacC – Phase C current

Weather Stations (WS)

WindSpeed

WindDirection

Temp1 – Module temperature

Temp2 – Secondary module temperature

TempF – Ambient temperature

Sun – Irradiance

Sun2 – Irradiance (GHI)

Nip – Normal Incidence Pyrheliometer

Rain – Rain sensor

Snow – Snow depth

RH – Relative humidity

Bar – Barometric pressure

CabF – Cabinet temperature

DC strings and zones (SA, SC, SS)

ZoneAmps# – Combiner output current for channel #.

I# - String output current for channel #

Thermal meters (TS, TP)

BTU1 – Energy

BtuPerHr – Power

GPM – Flow rate

ReturnF – Return temperature

SupplyF – Supply temperature

Site Summary (SU)

InvKWH – Cumulative production (inverter data)

ProdKWH – Cumulative production (from production meters)

Insolation – Wh/m^2

ModuleTemp - °C