



SolarEdge Monitoring Server API

Last update: May 2016

SolarEdge API

Contents

| | |
|---|----------|
| SolarEdge API | 2 |
| Contents | 2 |
| General | 4 |
| Purpose and scope..... | 4 |
| Acronyms and abbreviations | 4 |
| Introduction | 4 |
| Technical Information | 5 |
| Security | 5 |
| Language and Time Encoding | 6 |
| Request Format | 6 |
| Response Formats | 6 |
| JSONP Support | 6 |
| Error Handling..... | 7 |
| Usage Limitations | 7 |
| Specific API Usage Limitations | 7 |
| Daily Limitation | 7 |
| Concurrency Limitation | 7 |
| Bulk Use | 7 |
| API Description | 8 |
| Site Data API | 8 |
| Site List..... | 8 |
| Site Details | 9 |
| Site Data: Start and End Dates..... | 10 |
| Site Data: Bulk Version | 10 |
| Site Energy | 11 |
| Site Energy: Bulk Version | 11 |
| Site Energy – Time Period..... | 12 |
| Site Energy – Time Period: Bulk Version | 12 |
| Site Power | 13 |
| Site Power: Bulk version | 14 |
| Site Overview..... | 15 |
| Site Overview: Bulk Version | 15 |
| Site Power - Detailed | 16 |
| Site Energy - Detailed | 18 |
| Site Power Flow | 19 |
| Storage Information | 21 |
| Site Image | 23 |
| Installer Logo Image | 24 |
| Site Equipment API | 25 |
| Components List | 25 |
| Inventory | 25 |
| Inverter Technical Data..... | 27 |
| Account List API | 30 |
| Sensors API | 31 |
| Get Sensor List | 31 |

| | |
|-------------------------|-----------|
| Get Sensor Data | 32 |
| API Versions | 34 |
| Current Version | 34 |
| Supported Version | 34 |
| Data Types | 35 |
| Time Unit | 35 |
| Site Status | 35 |
| Site Type | 35 |

General

Purpose and scope

The purpose of this document is to outline the Application Programming Interface (API) available via SolarEdge Cloud-Based Monitoring Platform.

The web services allow access to data saved in the monitoring server while keeping the data secured for authorized users.

This document provides information about the technical features of the API, and describes each API with its parameters formats and other details.

Acronyms and abbreviations

The following table lists acronyms used in this document

| Abbreviation | Meaning |
|--------------|----------------------------------|
| API | Application Programing Interface |
| WS | Web Services |
| REST | Representational State Transfer |
| CSV | Comma Separated Values |
| JSON | JavaScript Object Notation |
| XML | Extensible Mark-up Language |

Introduction

The SolarEdge API allows other software applications to access its monitoring system database for data analysis purposes, fleet management, displaying system data in other applications, etc.

The following is a list of available APIs:

| API Name | API Output |
|--------------------------------|---|
| Site List | A list of sites for a given account, with the information on each site. This list allows convenient search, sort and pagination |
| Site Details | Details of a chosen site |
| Site Data: Start and End Dates | The site energy production start and end dates |
| Site Energy | Site energy measurements |
| Site Energy – Time Period | Site energy for a requested timeframe |
| Site Power | Site power measurements in a 15-minute resolution |
| Site Overview | Site current power, energy production (today, this month, lifetime) and lifetime revenue |
| Site Power | Detailed site power measurements including meters such as consumption, export (feed-in), import (purchase), etc. |
| Site Energy | Detailed site energy measurements including meters such as consumption, export (feed-in), import (purchase), etc. |
| Site Power Flow | Get the power flowchart of the site |
| Storage | Get detailed storage information from batteries including the state of energy, power and lifetime energy |
| Site Image | The site image as uploaded to the server, either scaled or original size |
| Installer Logo Image | The installer logo image as uploaded to the server. |
| Components List | List of inverters with name, model, manufacturer, serial number and status |
| Inventory | Information about the SolarEdge equipment, including: inverters/SMIs, batteries, meters, gateways and sensors |
| Inverter Technical Data | Technical data on the inverter performance for a requested period of time |

| | |
|------------------|---|
| Account List API | The account details and list of all sub-accounts |
| Get Sensor List | The list of sensors installed in the site |
| Get Sensor Data | The measurements of the sensors installed in the site |
| API Versions | The current and supported version numbers |

An API key must be used in order to submit API requests.

Account users should generate an Account Level API Key, and system owners should generate a Site level API Key.

The API link is <https://monitoringapi.solaredge.com>

Please review the API terms and conditions at this link: <http://monitoring.solaredge.com/solaredge-web/termsAndConditionsAPI.html>

Display requirements:

When displaying information from the API, place the SolarEdge logo where it is clear to the user that the information source is SolarEdge's monitoring system.

The logo should link to <http://solaredge.com>

For any assistance with display, send an email to support@solaredge.com

Technical Information

The SolarEdge API is built as RESTful service:

- It uses predictable, resource oriented URLs
- It has built-in HTTP capabilities for passing parameters via the API
- It responds with standard HTTP codes
- It can return results in XML, JSON (including JSONP support) or CSV format.

Security

- The API can be accessed via HTTPS protocol only. SolarEdge monitoring server supports both HTTPS/1.0 and HTTPS/1.1 protocols.
- All APIs are secured via an access token: every access to the API requires a valid token as a URL parameter named `api_key`. For example: `api_key=L4QLVQ1LOKQX2193VSEICXW61NP6B1O`

► To generate an account users API key:

In the Account Admin > Company Details tab > API Access section:

- 1 Acknowledge reading and agreeing to the SolarEdge API Terms & Conditions.
- 2 Click Generate API key.
- 3 Copy the key.
- 4 Click **Save**.
- 5 Use the key in all API requests

► To generate a Site API key:

In the Site Admin > Site Details tab > API Access section:

- 1 Acknowledge reading and agreeing to the SolarEdge API Terms & Conditions.
- 2 Click Generate API key.
- 3 Copy the key.
- 4 Click **Save**.
- 5 Use the key in all API requests

Language and Time Encoding

- When using special characters or spaces in URL, they must be **url encoded**.
- The monitoring server data can be in different languages therefore data is retrieved using UTF-8.
- Date and time formats in all APIs are:
 - Date and time: YYYY-MM-DD hh:mm:ss
 - Date only: YYYY-MM-DD
- Dates are always in the time zone of the site.
- All physical measures are in the metric units system.
- Temperature values are always in Celsius degrees.

Request Format

The request format and parameters are specified per each API, and conform to the HTTP and REST standards.

Parameter order in the request is not significant.

Response Formats

The user can request data in the following formats:

- JSON (application/json)
- XML(application/xml)
- CSV (text/csv).

See specific APIs in the next sections for supported format in each API.

If a specific format is not requested, the data will be retrieved via JSON. If the requested format is invalid, the system will generate HTTP error "Media not supported".

The API user can request the response format in one of the methods below. The system handles the response format as follows:

- URL format Parameter – ...&format=application/json
- Path extension – the name of the API will be followed by the requested format name ../details.json&...
- HTTP header – based on 'Accept' header e.g. Accept application/json

JSONP Support

The API supports JSONP calls by appending a callback parameter with the name of a callback method at the end of the request.

As JSONP content type is application/javascript, make sure the client sends this content type in the Accept header, otherwise HTTP 406 error may occur.

The following example shows a JSON call VS. JSONP call:

■ JSON:

- Request:

```
http://monitoringapi.solaredge.com/1/details.json?api_key=[your_api_key]
```

- Response body:

```
{"details":{..}}
```

■ JSONP:

- Request:

```
http://monitoringapi.solaredge.com/1/details.json?api_key=[your_api_key]&callback=myFunction
```

- Response body:

```
myFunction({"details":{..}});
```

Error Handling

The API system uses standard HTTP error codes for reporting errors that occurred while calling the API. The monitoring server API supports standard HTTP error codes, for example: if the user access is of an unknown resource, an HTTP 404 error will be returned.

Usage Limitations

Usage limitations are enforced to prevent abuse of the monitoring server API, and these limitations may be changed in the future without notice. Additionally, a request rate limit is applied to prevent abuse of the service. If you exceed the limitations, an error message appears in the monitoring server API. If the limitation is further exceeded, the system may temporarily be nonoperational, or your access to the monitoring server API may be blocked.

Specific API Usage Limitations

Specific APIs may enforce different usage limitations based on parameters sent by the client. Refer to the next sections for details on specific API usage limitations.

If there is a violation of a specific API validation, the `HTTP 403 - forbidden` status code is returned.

Daily Limitation

Use of the monitoring server API is subject to a query limit of total 300 requests for a specific account token and a specific site ID from the same source IP.

APIs that do not have a specific ID (e.g. Site List, Account List) will be counted as part of the total query limit.

For example, a user can execute 10 daily site list (account level) API calls, and then 290 API calls for every site in the account (e.g. 290 calls using the account token for "siteid 12345" and 290 calls using the account token for "siteid 12346").

An additional request to site 12345 or 12346 will result in `HTTP 429 error - too many requests..`

Concurrency Limitation

The monitoring server API allows up to 3 concurrent API calls from the same source IP. Any additional concurrent calls will return `HTTP 429 error - too many requests.`

To execute APIs concurrently without exceeding the above limitation, it is the client responsibility to implement a throttling mechanism on the client side.

Bulk Use

Some of the APIs offer a bulk form, that is, they take multiple site IDs as an input. The list is separated by a comma and should contain up to 100 site IDs.

A bulk call for multiple sites consumes 1 call from the daily quota allowed for each of the sites included in the call.

API Description

Site Data API

Site List

Returns a list of sites related to the given token. This API accepts parameters for convenient search, sort and pagination.

URL: `/sites/list`

Example URL (with all options):

`https://monitoringapi.solaredge.com/sites/list?size=5&searchText=Lyon&sortProperty=name&sortOrder=ASC`

Method: GET

Formats: JSON, XML and CSV

- **Request:** The following are parameters to include in the request.

| Parameter | Type | Mandatory | Default Value | Description |
|--------------|---------|-----------|----------------|--|
| size | Integer | No | 100 | The maximum number of sites returned by this call. The maximum number of sites that can be returned by this call is 100. If you have more than 100 sites, just request another 100 sites with startIndex=100. This will fetch sites 100-199. |
| startIndex | Integer | No | 0 | The first site index to be returned in the results |
| searchText | String | No | | Search text for this site. Searchable site properties: <ul style="list-style-type: none"> ▪ Name ▪ Notes ▪ Address ▪ City ▪ Zip code ▪ Full address ▪ Country |
| sortProperty | String | No | | A sorting option for this site list, based on one of its properties. Available sort properties: <ul style="list-style-type: none"> ▪ Name – sort by site name ▪ Country – sort by site country ▪ State – sort by site state ▪ City – sort by site city ▪ Address – sort by site address ▪ Zip – sort by site zip code ▪ Status – sort by site status ▪ PeakPower – sort by peak power ▪ InstallationDate – sort by installation date ▪ Amount – sort by amount of alerts ▪ MaxSeverity – sort by alert severity ▪ CreationTime – sort by site creation time |
| sortOrder | String | No | ASC | Sort order for the sort property. Allowed values are ASC (ascending) and DESC (descending). |
| Status | String | No | Active,Pending | Select the sites to be included in the list by their status: <ul style="list-style-type: none"> ▪ Active ▪ Pending ▪ Disabled ▪ All Default list will include Active and Pending sites. |

- **Response:** The returned data is the site list, including the sites that match the given search criteria. For each entry, the following information is displayed:
 - id - the site ID
 - name - the site name
 - account id -the account this site belongs to
 - status - the site status (see *Site Status* on page 35)
 - peak power - site peak power
 - CURRENCY
 - installationDate – site installation date (format: yyyy-MM-DD hh:mm:ss)
 - notes
 - type – site type (see *Site Type* on page 35)
 - location - includes country, state, city, address, secondary address, time zone and zip.
 - alertQuantity - number of open alerts this site have.
 - alertSeverity - the highest alert severity this site have
 - publicSettings - includes if this site is public and its public name
- **Example:** JSON output:

```
{
  "Sites": {
    "count": 1567,
    "list": [
      {
        "id": 1,
        "name": "Test",
        "accountId": 0,
        "status": "Active",
        "peakPower": 10.0,
        "currency": "EUR",
        "installationDate": "2012-06-08 00:00:00",
        "notes": "test notes",
        "type": "Optimizers & Inverters",
        "location": {
          "country": "the country",
          "state": "the state",
          "city": "the city",
          "address": "the address",
          "address2": "the address 2",
          "zip": "00000",
          "timeZone": "GMT"
        },
        "alertQuantity": 0,
        "alertSeverity": "NONE",
        "uris": {
          "PUBLIC_URL": "the public URL name",
          "IMAGE_URI": "the site image link"
        },
        "publicSettings": {
          "name": "the public name",
          "isPublic": true
        }
      },
      {
        "id": 1,
        "name": "Test",
        "accountId": 0,
        "status": "Active",
        "peakPower": 10.0,
        "currency": "EUR",
        "installationDate": "2012-06-08 00:00:00",
        "notes": "test notes",
        "type": "Optimizers & Inverters",
        "location": {
          "country": "the country",
          "state": "the state",
          "city": "the city",
          "address": "the address",
          "address2": "the address 2",
          "zip": "00000",
          "timeZone": "GMT"
        },
        "alertQuantity": 0,
        "alertSeverity": "NONE",
        "uris": {
          "PUBLIC_URL": "the public URL name",
          "IMAGE_URI": "the site image link"
        },
        "publicSettings": {
          "name": "the public name",
          "isPublic": true
        }
      }
    ]
  }
}
```

Site Details

Displays the site details, such as name, location, status, etc.

URL: `/site/{siteId}/details`

Example URL: <https://monitoringapi.solaredge.com/site/1/details>

Method: GET

Accepted formats: JSON and XML

- **Request:** The following parameter is included in the request:

| Parameter | Type | Mandatory | description |
|-----------|---------|-----------|---------------------|
| siteId | Integer | Yes | The site identifier |

- **Response:** The returned data includes the following site information:
 - id – the site ID
 - name – the site name
 - account id – the account this site belongs to
 - status – the site status (see *Site Status* on page 35)
 - peak power – site peak power
 - currency
 - installationDate – site installation date (format: yyyy-MM-DD hh:mm:ss)
 - notes
 - type – site type (see *Site Type* on page 35)

- location - includes country, state, city, address, secondary address, zip and time zone.
- alertQuantity - number of open alerts this site have.
- alertSeverity – the highest alert severity this site have
- publicSettings - includes if this site is public (isPublic) and its public name (name)

■ **Example:** JSON output:

```
{ "details": { "id": 0, "name": "site
name", "accountId": 0, "status": "Active", "peakPower": 9.8, "currency": "EUR", "installationDate": "2012-08-16
00:00:00", "notes": "my notes", "type": "Optimizers & Inverters", "location": { "country": "my country", "state": "my
state", "city": "my city", "address": "my address", "address2": "", "zip": "0000",
"timeZone": "GMT", "alertQuantity": 0, "alertSeverity": "NONE", "uris": { "IMAGE_URI": "site image
uri"}, "publicSettings": { "name": null, "isPublic": false } } }
```

Site Data: Start and End Dates

Description: Return the energy production start and end dates of the site.

URL: `/site/{siteId}/dataPeriod`

Example URL: <https://monitoringapi.solaredge.com/site/1/dataPeriod>

Method: GET

Accepted formats: JSON and XML

- **Request:** The following parameter is included in the request:

| Parameter | Type | Mandatory | description |
|-----------|---------|-----------|---------------------|
| siteId | Integer | Yes | The site identifier |

- **Response:** The returned data includes <start, end> dates of the requested site. In case this site is not transmitting, the response is “null”.

- **Example:** JSON output example:

```
{ "dataPeriod": { "startDate": "2013-05-05 12:00:00", "endDate": "2013-05-28 23:59:59" } }
```

For non-transmitting sites:

```
{ "dataPeriod": { "startDate": null, "endDate": null } }
```

Site Data: Bulk Version

This section describes the use of the above API for a bulk call.

URL: `/sites/{siteId 1},{siteId 2},...,{siteId n}/dataPeriod`

Example URL: <https://monitoringapi.solaredge.com/sites/1,4/dataPeriod>

- **Response:** The response includes a list of <start, end> data transmission dates for the requested sites.

The value “null” will be displayed for sites that have no data (not transmitting).

Note that if the list contains site IDs for which the user has no permission to view, the system will generate a *403 Forbidden* error with a proper description.

- **Example:** JSON format example:

In this response, the sample site 4 is not transmitting.

```
{
  "dataPeriod": {
    "count": 2,
    "list": [
      {
        "id": 1,
        "startDate": "2013-05-05 12:00:00",
        "endDate": "2013-05-28 23:59:59"
      },
      {
        "id": 4,
        "startDate": null,
        "endDate": null
      }
    ]
  }
}
```

Site Energy

Description: Return the site energy measurements.

URL: `/site/{siteId}/energy`

Example URL: <https://monitoringapi.solaredge.com/site/1/energy?timeUnit=DAY&endDate=2013-05-30&startDate=2013-05-01>

Method: GET

Accepted formats: JSON, XML and CSV

- **Usage limitation:** This API is limited to one year when using `timeUnit=DAY` (i.e., daily resolution) and to one month when using `timeUnit=QUARTER_OF_AN_HOUR` or `timeUnit=HOUR`. This means that the period between `endTime` and `startTime` should not exceed one year or one month respectively. If the period is longer, the system will generate error 403 with proper description.
- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|--|
| siteId | Integer | Yes | The site identifier |
| startDate | String | Yes | The start date to return energy measurement |
| endDate | String | Yes | The end date return energy measurement |
| timeUnit | String | | Aggregation granularity, see <i>Time Unit</i> on page 35. Default : DAY. Allowed values are: QUARTER_OF_AN_HOUR, HOUR, DAY, WEEK, MONTH, YEAR |

- **Response:** The response includes the requested time unit, the units of measurement (e.g. Wh), and the pairs of date and energy for every date (`{"date": "2013-06-01 00:00:00", "value": null}`). The date is calculated based on the time zone of the site. "null" means there is no data for that time.
- **Example:** JSON output:

```
{ "energy": { "timeUnit": "DAY", "unit": "Wh", "values": [ { "date": "2013-06-01 00:00:00", "value": null }, { "date": "2013-06-02 00:00:00", "value": null }, { "date": "2013-06-03 00:00:00", "value": null }, { "date": "2013-06-04 00:00:00", "value": 67313.24 } ] } }
```

Site Energy: Bulk Version

This section describes the use of that the above API for a bulk call.

URL: `/sites/{siteId 1},{siteId 2},...,{siteId n}/energy`

Example URL: <https://monitoringapi.solaredge.com/sites/1,4/energy?timeUnit=DAY&startDate=2013-05-01&endDate=2013-05-30>

- **Response:** The response includes the requested time unit, the units of measurement (e.g. Wh), and the list of sites. For each site there is a list of items which include a time stamp and the energy produced in that period. Example: (`{"date": "2013-06-01 00:00:00", "value": 1500.12}`). The date is calculated based on the time zone of every site; if there is no value for the selected time, "null" will be displayed for the value.
Note that if the list contains site IDs for which the user has no permission to view, the system will generate a 403 *Forbidden* error with a proper description.

- **Example:** JSON format example:

```
{
  "energy": {
    "timeUnit": "DAY",
    "unit": "Wh",
    "count": 2,
    "list": [
      {
        "id": 1,
        "values": [
          {
            "date": "2013-06-01 00:00:00",
            "value": null
          },
          {
            "date": "2013-06-02 00:00:00",
            "value": null
          }
        ]
      },
      {
        "id": 4,
        "values": [
          {
            "date": "2013-06-01 00:00:00",
            "value": null
          },
          {
            "date": "2013-06-02 00:00:00",
            "value": null
          }
        ]
      }
    ]
  }
}
```

```

        "date": "2013-06-03 00:00:00",
        "value": null
      },
      {
        "date": "2013-06-04 00:00:00",
        "value": 67313.24
      }
    ]
  },
  {
    "id": 4,
    "values": [
      {
        "date": "2013-06-01 00:00:00",
        "value": null
      },
      {
        "date": "2013-06-02 00:00:00",
        "value": null
      },
      {
        "date": "2013-06-03 00:00:00",
        "value": null
      },
      {
        "date": "2013-06-04 00:00:00",
        "value": 67313.24
      }
    ]
  }
]
}

```

Site Energy – Time Period

Description: Return the site total energy produced for a given period.

URL: `/site/{siteId}/timeFrameEnergy`

Example URL: <https://monitoringapi.solaredge.com/site/1/timeFrameEnergy?startDate=2013-05-01&endDate=2013-05-06>

Method: GET

Accepted formats: JSON and XML

- **Usage limitation:** This API is limited to one year when using timeUnit=DAY (i.e., daily resolution). This means that the period between endTime and startTime should not exceed one year. If the period is longer, the system will generate error 403 with proper description
- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---|
| siteId | Integer | Yes | The site identifier |
| startDate | String | Yes | The start date to calculate energy generation |
| endDate | String | Yes | The end date to calculate energy generation |

- **Response:** The response includes the energy summary for the given time period with units of measurement (e.g. Wh) The date is calculated based on the time zone where the site is located.
- **Example:** JSON output:

```
{"timeFrameEnergy": {"energy": 761985.8, "unit": "Wh"}}
```

Site Energy – Time Period: Bulk Version

This section describes the use of that the above API for a bulk call.

URL: `/sites/{siteId 1},{siteId 2},...,{siteId n}/timeFrameEnergy`

Example URL: <https://monitoringapi.solaredge.com/sites/1,4/energy?timeFrameEnergy?startDate=2013-05-01&endDate=2013-05-06>

- **Response:** The response includes the units of measurement (e.g. Wh), and the list of sites that include energy summary for the given time period.

The date is calculated based on the time zone of every site; if no data exists for the requested period, "null" will be displayed for the value field.

Note that if the list contains site IDs for which the user has no permission to view, the system will generate a *403 Forbidden* error with a proper description.

- **Example:** JSON format example:

```
{
  {
    "timeFrameEnergy": {
      "unit": "Wh",
      "count": 4,
      "list": [{
        "id": 1,
        "energy": 761985.8
      },
      {
        "id": 4,
        "energy": 234284.4
      },
      {
        "id": 534,
        "energy": null
      },
      {
        "id": 222,
        "energy": 9984724.5
      }
    ]
  }
}
```

Site Power

Description: Return the site power measurements in 15 minutes resolution.

URL: `/site/{siteId}/power`

Example URL: `https://monitoringapi.solaredge.com/site/1/power?startTime=2013-05-5%2011:00:00&endTime=2013-05-05%2013:00:00`

Method: GET

Accepted formats: JSON, XML and CSV

- **Usage limitation:** This API is limited to one-month period. This means that the period between `endTime` and `startTime` should not exceed one month. If the period is longer, the system will generate error 403 with proper description.
- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|--|
| siteId | Integer | Yes | The site identifier |
| startTime | String | Yes | The start (date + time) to get power measurements. |
| endTime | String | Yes | The end (date + time) to get power measurements |

- **Response:** The response includes the time unit (i.e. `QUARTER_OF_AN_HOUR`), the measurement units (e.g. Watt) and the pairs of date and power (in Watts) for every date (`{"date":"2013-06-04 14:00:00","value":7722.3896}`)
The date is calculated in ticks starting 1-1-1970 and presented based on the time zone of the site. "null" means there is no data for that time.
- **Example:** JSON output:

```
{"power":{"timeUnit":"QUARTER_OF_AN_HOUR","unit":"W","values":[{"date":"2013-06-04
11:00:00","value":7987.03}, {"date":"2013-06-04 11:15:00","value":9710.121}, {"date":"2013-06-04
11:30:00","value":8803.309}, {"date":"2013-06-04 11:45:00","value":9000.743}, {"date":"2013-06-04
12:00:00","value":6492.2075}, {"date":"2013-06-04 12:15:00","value":7395.716}, {"date":"2013-06-04
12:30:00","value":8855.878}, {"date":"2013-06-04 12:45:00","value":6551.6655}, {"date":"2013-06-04
```

```
13:00:00", "value": 8114.938}, {"date": "2013-06-04 13:15:00", "value": 7466.171}, {"date": "2013-06-04
13:30:00", "value": 6595.561}, {"date": "2013-06-04 13:45:00", "value": 8824.195}, {"date": "2013-06-04
14:00:00", "value": 7722.3896}]}}
```

Site Power: Bulk version

This section describes the use of that the above API for a bulk call.

URL: `/sites/{siteId 1},{siteId 2},...,{siteId n}/power`

Example URL: `https://monitoringapi.solaredge.com/sites/1,4/power?startTime=2013-06-04 11:00:00&endTime=2013-06-04 14:00:00`

- Response:** The response includes the resolution of time measurements (e.g. QUARTER_OF_AN_HOUR), units of measurement (e.g. W), and the list of sites that include date and power in the given resolution. The date is calculated in ticks starting 1-1-1970 and presented based on the time zone of the site. If no data exists for the requested period, "null" will be displayed for the value field. Note that if the list contains site IDs for which the user has no permission to view, the system will generate a *403 Forbidden* error with a proper description.
- Example:** JSON format example:

```
{
  "power": {
    "timeUnit": "QUARTER_OF_AN_HOUR",
    "unit": "W",
    "count": 2,
    "list": [{
      "id": 1,
      "values": [{
        "date": "2013-06-04 11:00:00",
        "value": 7987.03
      },
      {
        "date": "2013-06-04 11:15:00",
        "value": 9710.121
      },
      {
        "date": "2013-06-04 11:30:00",
        "value": 8803.309
      },
      {
        "date": "2013-06-04 11:45:00",
        "value": 9000.743
      },
      {
        "date": "2013-06-04 12:00:00",
        "value": 6492.207
      }
    ]
    },
    {
      "id": 4,
      "values": [{
        "date": "2013-06-04 11:00:00",
        "value": 7987.03
      },
      {
        "date": "2013-06-04 11:15:00",
        "value": 9710.121
      },
      {
        "date": "2013-06-04 11:30:00",
        "value": 8803.309
      },
      {
        "date": "2013-06-04 11:45:00",
        "value": 9000.743
      },
      {
        "date": "2013-06-04 12:00:00",
        "value": 6492.2075
      }
    ]
  }
}
```

Site Overview

Description: Display the site overview data.

URL: `/site/{siteId}/overview`

Example URL: <https://monitoringapi.solaredge.com/site/{siteId}/overview>

Method: GET

Accepted formats: JSON and XML

- **Request:** The following parameter is included in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---------------------|
| siteId | Integer | Yes | The site identifier |

- **Response:** The response includes the site current power, daily energy, monthly energy, yearly energy and life time energy.
- **Example:** JSON output:

```
{
  "overview": {
    "lastUpdateTime": "2013-10-01 02:37:47",
    "lifeTimeData": {
      "energy": 761985.75,
      "revenue": 946.13104
    },
    "lastYearData": {
      "energy": 761985.8,
      "revenue": 0.0
    },
    "lastMonthData": {
      "energy": 492736.7,
      "revenue": 0.0
    },
    "lastDayData": {
      "energy": 0.0,
      "revenue": 0.0
    },
    "currentPower": {
      "power": 0.0
    }
  }
}
```

Site Overview: Bulk Version

This section describes the use of that the above API for a bulk call.

URL: `/sites/{siteId 1},{siteId 2},...,{siteId n}/overview`

Example URL: <https://monitoringapi.solaredge.com/sites/1,4/overview>

- **Response:** The response includes the last update time, current power, and daily, monthly, yearly and life time energy and revenue measurements for each of the sites in the list
Note that if the list contains site IDs for which the user has no permission to view, the system will generate a *403 Forbidden* error with a proper description.

- **Example:** JSON format example:

```
{
  "overview": {
    "count": 2,
    "list": [
      {
        "id": 1,
        "lastUpdateTime": "2013-10-01 02:37:47",
        "lifeTimeData": {
          "energy": 761985.75,
          "revenue": 946.13104
        },
        "lastYearData": {
          "energy": 761985.8,
          "revenue": 0.0
        },
        "lastMonthData": {
          "energy": 492736.7,
          "revenue": 0.0
        },
        "lastDayData": {
          "energy": 0.0,
          "revenue": 0.0
        },
        "currentPower": {
          "power": 0.0
        }
      },
      {
        "id": 4,
        "lastUpdateTime": "2013-10-01 02:37:47",
        "lifeTimeData": {
          "energy": 761985.75,
          "revenue": 946.13104
        }
      }
    ]
  }
}
```

```

    "lastYearData": {
      "energy": 761985.8,
      "revenue": 0.0
    },
    "lastMonthData": {
      "energy": 492736.7,
      "revenue": 0.0
    },
    "lastDayData": {
      "energy": 0.0,
      "revenue": 0.0
    },
    "currentPower": {
      "power": 0.0
    }
  }
}
}
}

```

Site Power - Detailed

Detailed site power measurements from meters such as consumption, export (feed-in), import (purchase), etc.

Note: Calculated meter readings (also referred to as "virtual meters"), such as self-consumption, are calculated using the data measured by the meter and the inverters.

Base URL: `/site/{siteId}/powerDetails`

Method: GET

Accepted response formats: JSON (default), XML

- **Usage limitation:** This API is limited to one-month period. This means that the period between `endTime` and `startTime` should not exceed one month. If the period is longer, the system will generate error 403 with proper description.

■ Parameters

| Parameter | Type | Mandatory | Description |
|------------------------|--------|-----------|---|
| <code>siteId</code> | number | Yes | The site identifier |
| <code>startTime</code> | String | Yes | The power measured start time in yyyy-MM-DD hh:mm:ss format |
| <code>endTime</code> | String | Yes | The power measured end time in yyyy-MM-DD hh:mm:ss format |
| <code>meters</code> | String | No | Select specific meters only. If this value is omitted, all meter readings are returned. Value shall include entries from the following list separated by comma: <ul style="list-style-type: none"> ■ Production- AC production power meter / inverter production AC power (fallback) ■ Consumption - Consumption meter ■ SelfConsumption - virtual self-consumption (calculated) ■ FeedIn - Export to GRID meter ■ Purchased - Import power from GRID meter |

■ Example URL:

<https://monitoringapi.solaredge.com/site/1/powerDetails?meters=PRODUCTION,CONSUMPTION&startTime=2015-11-21%2011:00:00&endTime=2015-11-22%2013:00:00>

- **Response:** The response provides 15 minute resolution data series for each of the requested meters.

The response includes the following:

powerDetails - Root element

- **timeUnit** - The time unit of the data (i.e. QUARTER_OF_AN_HOUR)
- **unit** - Power measurement units (e.g. Watt)
- **meters** - List of meters. For each meter:
- **type** - The meter type (Production/Consumption/SelfConsumption/FeedIn (export)/Purchased (import))
- **values** - Pairs of date and power for every date
({"date":"2013-06-04 14:00:00", "value":7722.3896})
For dates in which no data exists the value will be committed

- Sample response: A sample response for a call such as:

<https://monitoringapi.solaredge.com/site/1/powerDetails?startTime=2015-11-21%2011:00:00&endTime=2015-11-21%2011:30:00>

```
{
  "powerDetails": {
    "timeUnit": "QUARTER_OF_AN_HOUR",
    "unit": "W",
    "meters": [
      {
        "type": "Consumption",
        "values": [
          {
            "date": "2015-11-21 11:00:00",
            "value": 619.8288
          },
          {
            "date": "2015-11-21 11:15:00",
            "value": 474.87576
          },
          {
            "date": "2015-11-21 11:30:00",
            "value": 404.7733
          }
        ]
      },
      {
        "type": "Purchased",
        "values": [
          {
            "date": "2015-11-21 11:00:00",
            "value": 619.8288
          },
          {
            "date": "2015-11-21 11:15:00",
            "value": 474.87576
          },
          {
            "date": "2015-11-21 11:30:00",
            "value": 404.7733
          }
        ]
      },
      {
        "type": "Production",
        "values": [
          {
            "date": "2015-11-21 11:00:00",
            "value": 0
          },
          {
            "date": "2015-11-21 11:15:00",
            "value": 0
          },
          {
            "date": "2015-11-21 11:30:00",
            "value": 0
          }
        ]
      },
      {
        "type": "SelfConsumption",
        "values": [
          {
            "date": "2015-11-21 11:00:00",
            "value": 0
          },
          {
            "date": "2015-11-21 11:15:00",
            "value": 0
          },
          {
            "date": "2015-11-21 11:30:00",
            "value": 0
          }
        ]
      },
      {
        "type": "FeedIn",
        "values": [
          {

```

```

        "date": "2015-11-21 11:00:00",
        "value": 0
      },
      {
        "date": "2015-11-21 11:15:00",
        "value": 0
      },
      {
        "date": "2015-11-21 11:30:00",
        "value": 0
      }
    ]
  }
}
}
}
}

```

Site Energy - Detailed

Detailed site energy measurements from meters such as consumption, export (feed-in), import (purchase), etc.

Note: Calculated meter readings (also referred to as "virtual meters"), such as self-consumption, are calculated using the data measured by the meter and the inverters.

Base URL: `/site/{siteId}/energyDetails`

Method: GET

Accepted response formats: JSON (default), XML

■ **Usage limitation:** This API is limited to:

- A year when using daily resolution (timeUnit=DAY)
- A month when using hourly resolution of higher (timeUnit=QUARTER_OF_AN_HOUR or timeUnit=HOUR)
- Lower resolutions (weekly, monthly, yearly) have no period limitation

In case the requested resolution is not allowed for the requested period, error 403 with proper description will be returned.

■ **Parameters:**

| Parameter | Type | Mandatory | Description |
|-----------|--------|-----------|---|
| siteId | number | Yes | The site identifier |
| startTime | String | Yes | The energy measured start time in yyyy-MM-DD hh:mm:ss format |
| endTime | String | Yes | The energy measured end time in yyyy-MM-DD hh:mm:ss format |
| timeUnit | String | No | Aggregation granularity. Default : DAY Allowed values are: <ul style="list-style-type: none"> ▪ QUARTER_OF_AN_HOUR ▪ HOUR ▪ DAY ▪ WEEK ▪ MONTH ▪ YEAR |
| meters | String | No | Select specific meters only. If this value is omitted, all meter readings are returned. Value shall include entries from the following list separated by comma: <ul style="list-style-type: none"> ▪ Production - Production energy meter ▪ Consumption - Consumption meter ▪ SelfConsumption - Virtual self-consumption (calculated) ▪ FeedIn - Export energy to GRID meter ▪ Purchased - Import energy from GRID meter |

■ **Example URL:**

<https://monitoringapi.solaredge.com/site/1/energyDetails?meters=PRODUCTION,CONSUMPTION&timeUnit=DAY&startTime=2013-05-15%2011:00:00&endTime=2013-05-25%2013:00:00>

■ **Response:** The response includes the following:

energyDetails - root element

- **timeUnit** - the requested time unit

- **unit** - The measurement units (e.g. Wh)
- **meters** - List of meters. For each meter:
- **type** - The meter type (Production/Consumption/SelfConsumption/FeedIn (export)/Purchased (import))
- **values** - Pairs of date and power for every date ({"date":"2013-06-04 14:00:00", "value":7722.3896}).
For dates in which no data exists the value will be committed (see sample)

■ **Sample response:** A sample response for a call such as:

<https://monitoringapi.solaredge.com/site/1/energyDetails?meters=PRODUCTION,CONSUMPTION&timeUnit=WEEK&startTime=2015-10-21%2011:00:00&endTime=2015-11-21%2011:30:00>

```
{
  "energyDetails": {
    "timeUnit": "WEEK",
    "unit": "Wh",
    "meters": [
      {
        "type": "Production",
        "values": [
          {
            "date": "2015-10-19 00:00:00"
          },
          {
            "date": "2015-10-26 00:00:00"
          },
          {
            "date": "2015-11-02 00:00:00"
          },
          {
            "date": "2015-11-09 00:00:00"
          },
          {
            "date": "2015-11-16 00:00:00",
            "value": 2953
          }
        ]
      },
      {
        "type": "Consumption",
        "values": [
          {
            "date": "2015-10-19 00:00:00"
          },
          {
            "date": "2015-10-26 00:00:00"
          },
          {
            "date": "2015-11-02 00:00:00"
          },
          {
            "date": "2015-11-09 00:00:00"
          },
          {
            "date": "2015-11-16 00:00:00",
            "value": 29885
          }
        ]
      }
    ]
  }
}
```

Site Power Flow

Retrieves the current power flow between all elements of the site including PV array, storage (battery), loads (consumption) and grid.

Note: Applies when export , import and consumption can be measured.

Base URL: </site/{siteId}/currentPowerFlow>

Method: GET

Accepted response formats: JSON (default), XML

- Parameters:

| Parameter | Type | Mandatory | Description |
|-----------|--------|-----------|---------------------|
| siteId | number | Yes | The site identifier |

- Example URL:** <https://monitoringapi.solaredge.com/site/1/currentPowerFlow>

- Response:** The response returns power flow for each of the elements in the system and their state.

In case the site does not support this information, the response should be an empty object.

Otherwise, the response includes the following:

- siteCurrentPowerFlow - Root element
- **unit** - The measurement units (e.g. Watt)
- **connections** - A table including all the relationships between the elements, and the power flow directions (producing element and consuming element)
- **from** element
- **to** element
- A list of elements - Element per entity type in the specific site
- **GRID** - always included in response
- **LOAD** - always included in response
- **PV** - included if the site has a PV array (measurement of PV produced power)
- **STORAGE** - included if the site has storage installed and enabled
- Parameters for each element:
 - For all included elements, the following parameters are provided:
 - **status** - The current status of the element (**Active / Idle / Disabled**)
 - **currentPower** - The current power of the element.
 - For **STORAGE**, negative power indicates the battery is discharging, while positive power indicates it is charging.
 - For **GRID**, negative power indicates purchase from the GRID, while positive power indicates feed-in to the GRID.
 - For Storage, the following additional properties are included:
 - **chargeLevel** - The current accumulated charge level of the storage (all batteries)
 - **critical**- If the accumulated storage charge level drops below a configurable level (currently 10%), this flag is returned
 - **timeLeft** - In Backup mode (**GRID is Disabled**), this property is returned to specify the time left before the storage energy runs out (estimated according to current load level).

- Sample response:

```
{
  "siteCurrentPowerFlow": {
    "unit": "W",
    "connections": [
      {
        "from": "GRID",
        "to": "Load"
      }
    ],
    "GRID": {
      "status": "Active",
      "currentPower": 3435.77978515625
    },
    "LOAD": {
      "status": "Active",
      "currentPower": 3435.77978515625
    },
    "PV": {
```

```

        "status": "Idle",
        "currentPower": 0
    },
    "STORAGE": {
        "status": "Idle",
        "currentPower": 0,
        "chargeLevel": 27,
        "critical": false
    }
}
}

```

Storage Information

Description: Get detailed storage information from batteries: the state of energy, power and lifetime energy.

Note: Applicable to systems with batteries.

Base URL: </site/{siteId}/storageData>

Method: GET

Accepted response formats: **JSON** (default), XML

- **Usage limitation:** This API is limited to one-week period.
Specifying a period that is longer than 7 days will generate error 403 with proper description.

- **Parameters:**

| Parameter | Type | Mandatory | Description |
|-----------|---------------------------------|-----------|---|
| siteId | number | Yes | The site identifier |
| startTime | String | Yes | Storage power measured start time in yyyy-MM-DD hh:mm:ss format |
| endTime | String | Yes | Storage power measured end time in yyyy-MM-DD hh:mm:ss format |
| serials | Comma separated list of Strings | No | Return data only for specific battery serial numbers; the list is comma separated. If omitted, the response includes all the batteries in the site |

- **Sample URL:** <https://monitoringapi.solaredge.com/site/1/storageData?serials=1111,2222&startTime=2015-05-22%2011:00:00&endTime=2015-05-25%2013:00:00>

- **Response:** The response includes the following:

- **storageData** - Root element
- **batteryCount** - Number of batteries included in the response
- **batteries** - A list of battery objects, each containing the following:
 - **serialNumber** - String - The battery serial number
 - **nameplate** - number - The nameplate (nominal) capacity of the battery
 - **telemetryCount** - number - The number of telemetries for this battery in the response
 - **telemetries** - A list of storage data telemetries. each entry contains:
 - **timeStamp** - String - Telemetry timestamp in the format of **YYY-MM-DD HH:MM:SS**
 - **power** - number - Positive power indicates the battery is charging, negative is discharging.
 - **batteryState** - number - can be one of the following: 0 (Invalid), 1 (Standby), 2 (Thermal Mgmt.), 3 (Enabled), 4 (Fault)
 - **lifeTimeEnergyCharged** - number - The lifetime energy charged into the battery in Wh.

- **Sample response:** A sample response for a call such as:

[https://monitoringapi.solaredge.com/site/1/storageData.json?startTime=2015-10-13 07:40:00&endTime=2015-10-13 11:40:00](https://monitoringapi.solaredge.com/site/1/storageData.json?startTime=2015-10-13%2007:40:00&endTime=2015-10-13%2011:40:00)

```
{
  "storageData": {
    "batteryCount": 1,
    "batteries": [
      {
        "nameplate": 1,
        "serialNumber": "BFA",
        "telemetryCount": 9,
        "telemetries": [
          {
            "timeStamp": "2015-10-13 08:00:00",
            "power": 12,
            "batteryState": 3,
            "lifeTimeEnergyCharged": 6
          },
          {
            "timeStamp": "2015-10-13 08:05:00",
            "power": 12,
            "batteryState": 3,
            "lifeTimeEnergyCharged": 6
          },
          {
            "timeStamp": "2015-10-13 08:10:00",
            "power": 12,
            "batteryState": 3,
            "lifeTimeEnergyCharged": 6
          },
          {
            "timeStamp": "2015-10-13 08:15:00",
            "power": 12,
            "batteryState": 3,
            "lifeTimeEnergyCharged": 6
          },
          {
            "timeStamp": "2015-10-13 08:20:00",
            "power": 12,
            "batteryState": 3,
            "lifeTimeEnergyCharged": 6
          },
          {
            "timeStamp": "2015-10-13 08:25:00",
```

```

    "power": 12,
    "batteryState": 3,
    "lifeTimeEnergyCharged": 6
  },
  {
    "timeStamp": "2015-10-13 08:30:00",
    "power": 12,
    "batteryState": 3,
    "lifeTimeEnergyCharged": 6
  },
  {
    "timeStamp": "2015-10-13 08:35:00",
    "power": 12,
    "batteryState": 3,
    "lifeTimeEnergyCharged": 6
  },
  {
    "timeStamp": "2015-10-13 08:40:00",
    "power": 12,
    "batteryState": 3,
    "lifeTimeEnergyCharged": 6
  }
]
}
]
}
}

```

Site Image

Description: Display the site image as uploaded by the user.

URL: `/site/{siteId}/siteImage/{name}`

Example URL: `https://monitoringapi.solaredge.com/site/1/siteImage/myname.jpg?hash=123456789`

Method: GET

- **Performance:** The image element returns with a hash element, which is consistent as long as the image is not changed. When executing the Site Image API while using the hash element, the server matches the image hash and the hash sent in the URL. If a match is found, the API returns an HTTP 304 code. In case the image hash that appears in the URL is different than the one stored in the server, the image will be downloaded. When using the `maxWidth` and `MaxHeight` parameters, the hash element will be ignored.
- **Image sizes:** By default, the API returns the same image that was uploaded to the monitoring portal. If an image in a different scale is required, the API supports it via the `maxWidth` and `maxHeight` parameters. The system will scale the image while keeping the aspect ratio of the original image, so the returned image will be smaller.
- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---|
| siteld | Integer | Yes | The site identifier |
| name | String | No | It is recommended to enter the site image name so the user can save the new image based on the name in the URL. |
| maxWidth | Integer | No | The maximum width to scale this image |
| maxHeight | Integer | No | The maximum height to scale this image |
| hash | Integer | No | The image hash |

- **Response:** The response includes one of the following:
 - The requested image
 - Error 404 – not found, if the site has no image
 - Error 304 – unmodified, if a hash attribute was specified

Installer Logo Image

Description: Return the site installer logo image as uploaded by the user. If such an image does not exist, the account installer logo is returned.

URL: `/site/{siteld}/installerImage/{name}`

Example URL: `https://monitoringapi.solaredge.com/site/1/installerImage/myname.jpg?hash=123456789`

Method: GET

- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---|
| siteld | Integer | Yes | The site identifier |
| name | String | No | It is recommended to enter the site image name so the user can save the new image based on the name in the URL. |

- **Response:** The response includes the requested image.

Site Equipment API

Components List

Description: Return a list of inverters/SMIs in the specific site.

URL: `/equipment/{siteId}/list`

Example URL (with all options): `https://monitoringapi.solaredge.com/equipment/2/list`

Method: GET

Accepted formats: JSON, XML and CSV

- **Request:** The following parameter is included in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---------------------|
| siteId | Integer | Yes | The site identifier |

- **Response:** The response includes a list of inverters/SMIs with their name, model, manufacturer and serial number..

- name – the inverter/SMI name
- manufacturer – the equipment manufacturer e.g. SolarEdge
- model – the inverter/SMI model e.g. SE16K
- serialNumber – the equipment short serial number

- **Example:** JSON output:

```
{
  "list": [
    {
      "name": "Inverter 1",
      "manufacturer": "SolarEdge",
      "model": "SE16K",
      "serialNumber": "12345678-00"
    },
    {
      "name": "Inverter 1",
      "manufacturer": "SolarEdge",
      "model": "SE16K",
      "serialNumber": "12345678-00"
    },
    {
      "name": "Inverter 1",
      "manufacturer": "SolarEdge",
      "model": "SE16K",
      "serialNumber": "12345678-00"
    },
    {
      "name": "Inverter 1",
      "manufacturer": "SolarEdge",
      "model": "SE16K",
      "serialNumber": "12345678-65"
    }
  ]
}
```

Inventory

Description: Return the inventory of SolarEdge equipment in the site, including: inverters/SMIs, batteries, meters, gateways and sensors.

URL: `/site/{siteId}/Inventory`

Example URL (with all options): `https://monitoringapi.solaredge.com/site/2/Inventory`

Method: GET

Accepted formats: JSON, XML

- **Request:** The following parameter is included in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---------------------|
| siteId | Integer | Yes | The site identifier |

- **Response:** The response includes a list equipment installed on site:

- Inverters – SolarEdge inverters
 - name – the inverter name e.g. Inverter 1
 - manufacturer – manufacturer name (SolarEdge)
 - model name e.g. SE16K
 - CPU Firmware version e.g. 2.52.311
 - DSP 1 Firmware version
 - DSP 2 Firmware version
 - communicationMethod – the communication interface used to connect to server. E.g. Ethernet.
 - serialNumber – the equipment serial number e.g. 7F123456-00

- connectedOptimizers – number of optimizers connected to the inverter
- thirdPartyInverters – 3rd party inverters
 - name – the inverter name, e.g.: Inverter 1
 - manufacturer – manufacturer name
 - model name
 - SN – serial number
- smiList – List of SMI devices
 - name – the inverter name e.g. Inverter 1
 - manufacturer – manufacturer name (SolarEdge)
 - model name e.g. SE16K
 - Firmware version e.g. 2.52.311
 - communicationMethod – the communication interface used to connect to server. e.g.: Ethernet.
 - serialNumber – the equipment serial number e.g.: 7F123456-00
 - connectedOptimizers – number of optimizers connected to the inverter
 - Meters
 - name – the inverter name e.g. “Feed In Meter”
 - Manufacturer – e.g. “WattNode”
 - Model – meter model number
 - SN – serial number (if applicable)
 - Type – meter type, e.g. “Production”
 - FirmwareVersion (if applicable)
 - ConnectedTo – Name of SolarEdge device the meter is connected to
 - connectedSolaredgeDeviceSN – serial number of the inverter / gateway the meter is connected to
 - form – physical for a HW meter or virtual if calculated by arithmetic between other meters
 - Sensors – Irradiance / wind / temperature sensors
 - connectedSolaredgeDeviceSN – the S/N of the device it is connected to e.g. 12345678-00
 - Id – e.g. “SensorDirectIrradiance”
 - connectedTo – name of the device it is connected to e.g. “Gateway 1”
 - Category – e.g. IRRADIANCE
 - Type – e.g. “Plane of array irradiance”
 - Gateways:
 - name – the inverter name e.g. Inverter 1
 - serialNumber – the equipment serial number e.g. 7F123456-00
 - Firmware version
 - Batteries
 - Name
 - Serial Number
 - Manufacturer - the battery manufacturer name
 - Model - the battery model name

- Nameplate capacity - the nameplate capacity of the battery as provided by the manufacturer
- Firmware version
- ConnectedTo – Name of SolarEdge device the battery is connected to
- connectedSolaredgeDeviceSN – serial number of the inverter / gateway the battery is connected to

■ **Example:** JSON output:

```
{
  "Inventory": {
    "meters": [
      {
        "name": "Production Meter",
        "manufacturer": "WattNode",
        "model": "WNC-3Y-480-MB",
        "firmwareVersion": "0013",
        "connectedSolaredgeDeviceSN": "12345678-00",
        "type": "Production",
        "form": "physical"
      }
    ],
    "sensors": [
      {
        "connectedSolaredgeDeviceSN": "12345678-00",
        "id": "SensorDirectIrradiance",
        "connectedTo": "Gateway 1",
        "category": "IRRADIANCE",
        "type": "Direct irradiance"
      },
      {
        "connectedSolaredgeDeviceSN": "12345678-00",
        "id": "SensorPlaneOfArrayIrradiance",
        "connectedTo": "Gateway 1",
        "category": "IRRADIANCE",
        "type": "Plane of array irradiance"
      }
    ],
    "gateways": [
      {
        "name": "Gateway 1",
        "firmwareVersion": "2.956.0",
        "SN": "12345678-00"
      }
    ],
    "batteries": [
      {
        "name": "Battery 1.1",
        "manufacturer": "NAME",
        "model": "10KWh",
        "firmwareVersion": "2.0",
        "connectedInverterSn": "12345678-01",
        "nameplateCapacity": 6400.0,
        "SN": "T123456789"
      }
    ],
    "inverters": [
      {
        "name": "Inverter 1",
        "manufacturer": "SolarEdge",
        "model": "SE20K",
        "firmwareVersion": "2.19.233",
        "SN": "12345678-01",
        "connectedOptimizers": 76
      },
      {
        "name": "Inverter 2",
        "manufacturer": "SolarEdge",
        "model": "SE20K",
        "firmwareVersion": "2.19.233",
        "SN": "12345678-02",
        "connectedOptimizers": 76
      }
    ]
  }
}
```

Inverter Technical Data

Description: Return specific inverter data for a given timeframe.

URL: `/equipment/{siteId}/{serialNumber}/data`

Example URL (with all options): `https://monitoringapi.solaredge.com/equipment/2/12345678-90/data?startTime=2013-05-5%2011:00:00&endTime=2013-05-05%2013:00:00`

Method: GET

Accepted formats: JSON, XML and CSV

- **Usage limitation:** This API is limited to one-week period. This means that the period between `endTime` and `startTime` should not exceed one week. If the period is longer, the system will generate error 403 with proper description.

- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|--------------|---------|-----------|--|
| siteId | Integer | Yes | The site identifier |
| serialNumber | String | Yes | The inverter short serial number |
| startTime | String | Yes | The start (date + time) to get inverter data |
| endTime | String | Yes | The end (date + time) to get inverter data |

- **Response:** The response includes technical parameters as for the inverter's performance (e.g., voltage, current, active power etc.), inverter type (1ph or 3ph), and software version. If an attribute is not supported based on the inverter version or type it will be omitted from the response.

| Original name | Comment | Data divided per phase |
|-----------------------|--|------------------------|
| timestamp | | no |
| Ac current | | yes |
| Ac voltage | | yes |
| AC frequency | | yes |
| QRef | | yes |
| CosPhi | | yes |
| Total Active Power | | no |
| apparentPower | Supported starting communication board version 2.474 | yes |
| activePower | Supported starting communication board version 2.474 | yes |
| reactivePower | Supported starting communication board version 2.474 | yes |
| Dc voltage | | no |
| groundFaultResistance | | no |
| powerLimit % | | no |
| Lifetime energy | Supported starting communication board version 2.474 | no |
| vL1ToN | | 1ph only |
| vL2ToN | | 1ph only |
| vL1To2 | | 3ph only |
| vL2To3 | | 3ph only |
| vL3To1 | | 3ph only |

■ Example:

```
{"data":{"count":9,"telemetries":[{"date":"2013-06-04
11:00:00","totalActivePower":null,"dcVoltage":46.9757,"L1Data":{"acCurrent":22.653,"acVoltage":11.6201,"acFr
equency":41.3468}},{"date":"2013-06-04
11:05:00","totalActivePower":null,"dcVoltage":238.514,"L1Data":{"acCurrent":9.76655,"acVoltage":19.0102,"acFr
equency":49.5571}},{"date":"2013-06-04
11:10:00","totalActivePower":null,"dcVoltage":195.84,"L1Data":{"acCurrent":11.5758,"acVoltage":16.8566,"acFr
equency":36.9939}},{"date":"2013-06-04
11:15:00","totalActivePower":null,"dcVoltage":394.716,"L1Data":{"acCurrent":21.4308,"acVoltage":20.7485,"acFr
equency":10.4653}},{"date":"2013-06-04
11:20:00","totalActivePower":null,"dcVoltage":378.997,"L1Data":{"acCurrent":5.50747,"acVoltage":7.48402,"acFr
equency":46.8013}},{"date":"2013-06-04
11:25:00","totalActivePower":null,"dcVoltage":45.9107,"L1Data":{"acCurrent":26.3345,"acVoltage":21.2422,"acFr
equency":21.1001}},{"date":"2013-06-04
11:30:00","totalActivePower":null,"dcVoltage":281.313,"L1Data":{"acCurrent":2.6314,"acVoltage":32.9308,"acFr
equency":26.677}},{"date":"2013-06-04
11:35:00","totalActivePower":null,"dcVoltage":45.9194,"L1Data":{"acCurrent":25.7302,"acVoltage":28.5508,"acFr
equency":3.33931}},{"date":"2013-06-04
11:40:00","totalActivePower":null,"dcVoltage":383.813,"L1Data":{"acCurrent":27.6644,"acVoltage":14.0232,"acFr
equency":32.8427}}]}
```

Account List API

Description: Return the account and list of sub-accounts related to the given token. This API accepts parameters for convenient search, sorting and pagination.

URL: /accounts/list

Example URL (with all options):

<https://monitoringapi.solaredge.com/accounts/list?size=5&searchText=someText&sortProperty=name&sortOrder=ASC>

Method: GET

Accepted formats: JSON, XML and CSV

- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Default Value | Description |
|--------------|---------|-----------|---------------|--|
| size | Integer | No | 100 | The maximum number of accounts returned by this call. If you have more than 100 sites, just request another 100 sites with startIndex=100. This will fetch sites 100-199. |
| startIndex | Integer | No | 0 | The first account index to be returned in the results |
| searchText | String | No | | Search text for this account. Searchable properties: <ul style="list-style-type: none"> ▪ Name – the account name ▪ Notes ▪ Email – contact person email ▪ Country ▪ State ▪ City ▪ Zip ▪ Full address |
| sortProperty | String | No | | A sorting option for this account list, based on one of its properties. Available sort properties: <ul style="list-style-type: none"> ▪ Name – sort by account name ▪ country – sort by account country ▪ city – sort by account city ▪ address – sort by account address ▪ zip – sort by account zip code ▪ fax – sort by account fax number ▪ phone – sort by account phone ▪ notes – sort by account notes |
| sortOrder | String | No | ASC | Sort order for the sort property. Allowed values are ASC (ascending) and DESC (descending). |

- **Response:** The returned data is the account data, including sub-accounts. For each entry, the following information is displayed:
 - id – account ID
 - name – account name
 - location – includes country, state, city, address, address2 (secondary address), zip
 - companyWebSite – the company web site
 - contactPerson – the account contact person first name and surname
 - email – the contact person email
 - phoneNumber – account phone number
 - faxNumber – account fax number
 - notes – account notes
 - parentId – account parent identifier

- **Example:** JSON output example:

```
{
  "accounts": {
    "count": 2638,
    "list": [
      {
        "id": 0,
        "name": " account 1",
        "location": {
          "country": "my country",
          "state": null,
          "city": null,
          "address": "my address 4",
          "address2": "my address 2",
          "zip": "00000"
        },
        "companyWebSite": "",
        "contactPerson": "Saar",
        "email": "mail@mail.com",
        "phoneNumber": "+00000000",
        "faxNumber": "",
        "notes": " ",
        "parentId": 32,
        "uris": null
      },
      {
        "id": 1,
        "name": " account 2",
        "location": {
          "country": "my country",
          "state": null,
          "city": null,
          "address": "my address 4",
          "address2": "my address 2",
          "zip": "00000"
        },
        "companyWebSite": "",
        "contactPerson": "Saar",
        "email": "mail@mail.com",
        "phoneNumber": "+00000000",
        "faxNumber": "",
        "notes": " ",
        "parentId": 32,
        "uris": null
      },
      {
        "id": 2,
        "name": " account 3",
        "location": {
          "country": "my country",
          "state": null,
          "city": null,
          "address": "my address 4",
          "address2": "my address 2",
          "zip": "00000"
        },
        "companyWebSite": "",
        "contactPerson": "Saar",
        "email": "mail@mail.com",
        "phoneNumber": "+00000000",
        "faxNumber": "",
        "notes": " ",
        "parentId": 32,
        "uris": null
      }
    ]
  }
}
```

Sensors API

Get Sensor List

Returns a list of all the sensors in the site, and the device to which they are connected.

Base URL: /equipment/{**siteId**}/sensors

Example: <https://monitoringapi.solaredge.com/equipment/2/sensors>

Method: GET

Accepted formats: JSON

- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|---------------------|
| siteId | Integer | Yes | The site identifier |

- **Response:** Returns the list of sensors installed in the site associated with the gateway they are connected with. Each entry will include the following parameters:

- **connectedTo:** name of the gateway the sensor is connected to
- **name:** the name of the sensor
- **measurement:** what the sensor measures, e.g.: SensorGlobalHorizontalIrradiance, SensorDiffusedIrradiance, SensorAmbientTemperature
- **type:** the sensor type e.g.: Temperature, Irradiance.
- **Example:** JSON output example:

```
{
  "SiteSensors": {
    "count": 3,
    "list": [
      {
        "connectedTo": "Gateway 19",
        "count": 2,
        "sensors": [
          {
            "name": "Global horizontal irradiance",
            "measurement": "SensorGlobalHorizontalIrradiance",
            "type": "IRRADIANCE"
          },
          {
            "name": "Diffused irradiance",
            "measurement": "SensorDiffusedIrradiance",
            "type": "IRRADIANCE"
          }
        ]
      }
    ]
  }
}
```

```

    ]]
  },
  {
    "connectedTo": "Gateway 1",
    "count": 1,
    "sensors": [{
      "name": "Ambient temperature",
      "measurement": "SensorAmbientTemperature",
      "type": "TEMPERATURE"
    }]
  }
}

```

Get Sensor Data

Returns the data of all the sensors in the site, by the gateway they are connected to.

Base URL: /site/{siteId}/sensors?{startDate}=<timestamp>&{endDate}=<timestamp>

Example: <https://monitoringapi.solaredge.com/site/2/sensors?startDate=2013-05-5%2011:00:00&endDate=2013-05-05%2013:00:00>

Method: GET

Accepted formats: JSON

- **Usage limitation:** This API is limited to one-week period. This means that the period between endDate and startDate should not exceed one week). If the period is longer, the system will generate error 403 with a description.
- **Request:** The following are parameters to include in the request:

| Parameter | Type | Mandatory | Description |
|-----------|---------|-----------|--|
| siteId | Integer | Yes | The site identifier |
| startDate | String | Yes | The start (date + time) to get sensor data |
| endDate | String | Yes | The end (date + time) to get sensor data |

- **Response:** Returns the telemetries reported by all sensors in the site, by the device they are connected to. Each entry will include the following parameters:
 - connectedTo: name of the gateway the sensor is connected to
 - count: the number of telemetries
 - date: timestamp of the telemetries
 - measurement: (e.g. ambientTemperature) and its numerical value (metric system)
- **Example:** JSON output example:

```

{
  "siteSensors": {
    "data": [{
      "connectedTo": "Gateway 19",
      "count": 0,
      "telemetries": []
    }],

```



```
{
  "connectedTo": "Gateway 1",
  "count": 427,
  "telemetries": [{
    "date": "2015-06-15 13:00:00",
    "ambientTemperature": -22.1155,
    "moduleTemperature": 47.2601,
    "windSpeed": 81.3652,
    ...
  }],
  {
    "connectedTo": "Gateway 3",
    "count": 427,
    "telemetries": [{
      "date": "2015-06-15 13:00:00",
      ...
    }],
    {
      "connectedTo": "Gateway 4",
      "count": 0,
      "telemetries": []
    },
    {
      "connectedTo": "Gateway 489",
      "count": 0,
      "telemetries": []
    },
    {
      "connectedTo": "Gateway 488",
      "count": 427,
      "telemetries": [{
        "date": "2015-06-15 13:00:00",
        ...
      }]
    }
  }
}
```

API Versions

As the monitoring API evolves over time, users of the monitoring API need to make sure their code is interacting with the formally supported version.

The monitoring API supports previous versions to some extent. This can be verified by executing the *Supported Version* API (see below).

The version format is <major.minor.release> where:

- **Major** - The main version number. This number increases when the version includes significant changes, which might not be backward compatible with previous versions in terms of APIs calls and returned results.
- **Minor** – The sub-version number. This number increases when the version includes some changes, which might not affect the APIs, however the returned results can contain more information than the previous minor version.
- **Release** – bug fixes

The user should optionally specify the version as a parameter for each API (except this API) e.g. version=1.0.0. If it is omitted, current version is assumed (see *Current Version* below).

Current Version

Description: Return the most updated version number in <major.minor.revision> format.

URL: */version/current*

Example URL: *https://monitoringapi.solaredge.com/version/current*

Method: GET

Accepted formats: JSON and XML

Method: GET

- **Request:** No parameters
- **Response:** The current version
- **Example:** JSON output:

```
{"version": "1.0.0"}
```

Supported Version

Description: Return a list of supported version numbers in <major.minor.revision> format.

URL: */version/supported*

Example URL: *https://monitoringapi.solaredge.com/version/current*

Method: GET

Accepted formats: JSON and XML

Method: GET

- **Request:** No parameters
- **Response:** A list of supported versions
- **Example:** JSON output:

```
{"supported": ["0.9.5", "1.0.0"]}
```

Data Types

Time Unit

Allowed values: DAY, WEEK, MONTH and YEAR

Site Status

- Active – The site is active
- Pending Communication – The site was created successfully however there is no communication yet from its inverters/SMI.

Site Type

- Optimizers and inverters
- Safety and monitoring interface
- Monitoring combiner boxes