

# Siderea PV Simulator

## REST API

# Contents

Introduction .....	3
Requirements/Skills .....	3
Using the Siderea REST API .....	3
getCalculationResult .....	4
getCalculationResult - the arrays parameter .....	8
array_layout .....	8
solar_module .....	10
solar_inverter .....	13
cable_dc .....	15
skyline.....	16
getSolarModules.....	17
getSolarInverters.....	18
getSolarArrayMountings .....	19
getMeteoStations .....	20
getDatesLatestMeteo.....	21
getDatesLatestHourly .....	22
getMaximumNumberOfArrays .....	23
getCredits.....	24
getThrottlingSettings.....	25
getVersions .....	26
Appendix A: API Errors.....	27

# Introduction

The Siderea REST API is a RESTful web service for communicating with the Siderea PV Simulator. This document provides the documentation for the available methods in this version of the Siderea REST API.

## Requirements/Skills

- Thorough understanding of solar power components and specifications.
- Beginner skills in PHP programming.
- Computer with internet access.
- Installed version of the PHP programming language.
- Terminal app.
- Simple code editor app.

## Using the Siderea REST API

The Siderea REST API uses standard HTTPS requests and returns data in JSON format. The API can be accessed through the following URL:

*<https://ionos.siderea.nl/pvsim/webservice/REST/>*

Each API method has its own path and additional parameters are passed to the API using POST.

Responses are returned in JSON format and include an 'error' element in case an error occurred. See *Appendix A: API Errors* for a complete list of all possible errors.

The Siderea REST API is freely accessible with **login** 'public' and **password** 'public' as POST variables for authentication. The API is rate limited; use the *getThrottlingSettings* method to get details about the current rate limiting settings.

A PHP client class (class.SidereaRESTAPI.php) and some ready-to-use example programs (GetYieldPVSystem.php, ListMeteostations.php) are available for [download here](#).

# getCalculationResult

This is the main API method used to perform a calculation for a PV system.

## URL

/getCalculationResult/

## Method

POST

## Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

**simulator\_version** string, one of the values returned by *getVersions*

"v9" - current stable version (C++ compiled library).

"v7" - deprecated (for legacy users). **DO NOT USE.**

Example:

```
"v9"
```

**start\_date** string with date in ISO 8601 format (yyyy-mm-dd)

Example:

```
"2019-01-01"
```

**end\_date** string with date in ISO 8601 format (yyyy-mm-dd)

Example:

```
"2023-12-31"
```

**latitude** float, decimal latitude, >=-60 and <=60

Example:

```
51.32
```

**longitude** float, decimal longitude,  $\geq -180$  and  $\leq 180$

Example:

```
5.69
```

**meteo\_stations** array, 0 to 2 ID's as returned by the `getMeteoStations` method

Example:

```
[235, 273]
```

*Note: if an empty array is passed to the function, the nearby MeteoStation based on latitude and longitude will be used for the calculations. If two ID's are provided, then the average of the meteorological data for the two stations will be used.*

**output\_type** string, one of: "days", "weeks", "months", "years"

Example:

```
"years"
```

**use\_hourly\_meteo** string, one of: "true", "false"

Example:

```
"true"
```

*Note: meteodata (global radiation and ambient temperature) is available as daily and hourly values. It is recommended to use hourly values (higher accuracy).*

**arrays** array, one-based array with incremental key for each array in the PV system.

Example:

```
{
  1: {
    "array_layout": {
      "azimuth": 134,
      "tilt": 38,
      "mounting": "on roof",
      "module_count": 12,
      "string_count": 1,
      "inverter_count": 1,
      "inverter_shared_with": 0,
      "power_limiter": 1500
    },
    "solar_module": {
      "id": 1
    },
    "solar_inverter": {
      "id": 1
    },
    "cable_dc": {
      "cable_length": 40,
      "cable_thickness": 4
    },
    "skyline": [ [5,10,5,5,10,5,5,10,5,5,10,5,5,10,5,5,10,5,5,10,5] ]
  },
  2: {
    "array_layout": {
      "azimuth": 134,
      ...etc...
    },
    ...etc...
  }
}
```

*Note: you can find more essential details about the "arrays" parameter in the section "getCalculationResult - the arrays parameter"*

## Success Response Example

```
{
  "allsky_horizontal_global": {
    "1991": 1016410.7656957,
    "1992": 1008111.2633572,
    "1993": 988365
  },
  "ac_energy_Wh": {
    "1991": 2877879.6430565,
    "1992": 2778925.5712883,
    "1993": 2724992.1603146
  },
  ...etc...
}
```

The available root-elements in the response are listed in the table below.

name	description
"allsky_horizontal_global"	Measured global horizontal radiation (Wh/m <sup>2</sup> )
"allsky_tilted_global"	Radiation on solar modules in (Wh/m <sup>2</sup> )
"allsky_tilted_global_refl"	Radiation on solar modules including reflection loss (Wh/m <sup>2</sup> )
"allsky_tilted_global_unshaded"	Radiation on solar modules without shading loss (Wh/m <sup>2</sup> )
"dc_energy_w_irrloss_Wh"	DC energy solar modules with irradiance loss (Wh)
"dc_energy_w_temploss_Wh"	DC energy solar modules with temperature loss (Wh)
"dc_energy_Wh"	DC energy solar modules (Wh)
"dc_cable_loss_Wh"	DC cable loss (Wh)
"ac_energy_Wh"	AC energy inverter(s) (Wh)
"tiltfactor"	Tilt Factor (tilted global radiation / horizontal global radiation)
"performance_ratio"	Performance Ratio (1 - system losses)

The format of the indexes within each element is determined by the chosen *output\_type*, see the table below.

output_type	index format in response	example
"days"	yyyy-mm-dd	"1991-12-29"
"weeks"	yyyy-ww	"1991-52"
"months"	yyyy-mm	"1991-12"
"years"	yyyy	"1991"

## Error Response Example

```
{
  "error": {
    "code":2021,
    "message":"Invalid azimuth for array_layout in Array 1"
  }
}
```

## getCalculationResult - the *arrays* parameter

The *arrays* parameter consists of a number of key-value pairs where the value describes one array of the PV system and the key assigns a number to this array. The minimum number of arrays in the *arrays* parameter is 1. The maximum allowed number of arrays can be retrieved using the *getMaximumNumberOfArrays* method. The key of the first array is 1 then incrementing by 1 for each other array.

Each value in the *arrays* parameter is in turn constructed from the following mandatory elements: *array\_layout*, *solar\_module*, *solar\_inverter*, *cable\_dc* and *skyline*.

### **array\_layout**

All of the following elements of *array\_layout* are mandatory.

***azimuth*** integer,  $\geq 0$  and  $\leq 360$ , 0 = North, 90 = East, 180 = South, 270 = West

Example:

```
180
```

***tilt*** integer,  $\geq 0$  and  $\leq 90$

Example:

```
30
```

***mounting*** string, one of the "string" values returned by *getSolarArrayMountings*

Example:

```
"on roof"
```

***module\_count*** integer,  $> 0$  and  $\leq 1000000$

Example:

```
12
```

***string\_count*** integer,  $> 0$  and  $\leq 1000000$

Example:

```
1
```

***inverter\_count*** integer,  $> 0$  and  $\leq 1000000$

Example:

```
1
```

**inverter\_shared\_with** integer,  $\geq 0$  and  $\leq$  the number of supplied arrays

Example:

```
0
```

*Note: this value is used to indicate whether or not this array is using a shared inverter with one or more other arrays. If the array is not sharing its inverter, set this value to 0. If the array does share its inverter, set this value to the key/number of the array it's sharing its inverter with. For example, if in a PV system with 3 arrays, arrays 1 and 2 are sharing an inverter, then set the values for inverter\_shared\_with as follows:*

*Array 1: inverter\_shared\_with: 2*

*Array 2: inverter\_shared\_with: 1*

*Array 3: inverter\_shared\_with: 0*

*Arrays that are sharing an inverter must have identical specifications for solar\_inverter.*

**powerlimiter (v9+ only)** optional, integer, =0 disabled,  $>0$  and  $\leq 1000000000$ , in watts

This option sets the ac-power grid feed-in limit imposed on all solar inverters in arrays where 'powerlimiter' is present. This option is ignored if the value exceeds the total ac-power of all solar inverters in arrays with 'powerlimiter' enabled. If 'powerlimiter' is set in multiple arrays the value must be equal (support for single powerlimiter only).

Example:

```
55000
```

*Note: powerlimiters allow for control of the grid feed-in power of connected solar inverters.*

## solar\_module

For the layout of the *solar\_module* component there are two options:

A known solar module referenced by *id*. A list of valid id's can be retrieved using the *getSolarModules* method.

Example:

```
"solar_module": {
  "id": 1
}
```

where:

**id** integer, a valid id returned by the *getSolarModules* method

Example:

```
4
```

ór

A custom solar module for which all required specifications are provided.

Example:

```
"solar_module": {
  "power_stc": 250,
  "voltage_mpp": 29.9,
  "current_mpp": 8.3612,
  "efficiency_stc": 0.155,
  "temp_coefficient": 0.0042,
  "absorption_coeff": 0.85,
  "efficiency_type": 0.045,
  "ar_coating": 1,
  "cable_length": 2,
  "cable_thickness": 4
}
```

*Note: if a valid id is set for solar\_module, the known specifications for this module will be used and any other set value will be ignored.*

**power\_stc** integer, >0 and <=1000, in watts

Example:

```
250
```

**voltage\_mpp** float, >0 and <=100, in volts

Example:

```
29.9
```

**current\_mpp** float, >0 and <=25, in ampere

Example:

8.3612

**efficiency\_stc** float, >0 and <=0.4

Example:

0.155

**temp\_coefficient** float, >0 and <=0.01

Example:

0.0042

**absorption\_coeff (v9+ only)** float, >0.5 and <1

The "absorption coefficient" is defined as the fraction of the solar module surface covered with solar cells.

Example:

0.85

*Note: datasheets of solar modules generally do not include the absorption coefficient. As a rule of thumb the absorption coefficient is proportional to power\_stc. Values range from 0.80 for modules of 100Wp or less to 0.90 for modules of 400Wp or more.*

**efficiency\_type** float, one of 0, 0.030, 0.035, 0.040, 0.045, 0.050, 0.060, 0.070, 0.080, 0.090, 0.100, 0.150 for version "v7" or >=0 and <=0.25 for other simulator versions. Floats are rounded to 3 decimals. The solar module efficiency\_type is defined as the reduction in efficiency (as a %) at 200W/m<sup>2</sup> relative to 1000W/m<sup>2</sup> where the efficiency at 1000W/m<sup>2</sup> is 1.

Example:

0.05

*Note: datasheets of solar modules generally do not include efficiencies at low irradiance . For most solar modules the relative efficiency reduction at 200W/m<sup>2</sup> is 0.05 (5%) or 0.07 (7%) for modules manufactured before 2010.*

**Anti-Reflective Coating (v9+ only)**

Refractive index of the anti-reflection coating of the solar-module cover.

*Specify if the module has an ar-coating or not*

**ar\_coating** integer, 1 or 0, 0 = without ar-coating, 1 = with ar-coating (a 'standard' refractive index of 1.38 is used).

Example:

1

ór

Specify the refractive index of the ar-coating

**ar\_coating** float, >1 and < 1.52, refractive index of ar-coating

Example:

1.36

**cable\_length** float, >0 and <=10, in meters

Example:

2.5

**cable\_thickness** float, >=1 and <=10, in mm<sup>2</sup>

Example:

4.5

*Note: cable\_length and cable\_thickness apply to the fixed cabling of the solar\_module itself. See also 'cable-dc' for specifying additional dc-cabling (to/from inverter)*

## solar\_inverter

For the layout of the *solar\_inverter* component there are two options:

A known solar inverter referenced by *id*. A list of valid id's can be retrieved using the *getSolarInverters* method.

Example:

```
"solar_inverter": {  
  "id": 1  
}
```

*id* integer, a valid id returned by the *getSolarInverters* method

Example:

```
3
```

ór

A custom solar module for which all required specifications are provided.

Example:

```
"solar_inverter": {  
  "power_ac_max": 3600,  
  "power_ac_limit": 3600,  
  "efficiency_max": 0.97,  
  "mpp_voltage_min": 175,  
  "mpp_voltage_max": 500,  
  "mppt_count ": 2,  
  "i_mppt_max": 23,  
  "i_dc_max": 0,  
  "with_transformer": 0,  
  "efficiency_type": 3010  
}
```

*Note: if a valid id is set for solar\_inverter, the known specifications for this inverter will be used and any other set value will be ignored.*

**power\_ac\_max** integer, >0 and <=1000000, in watts

Example:

```
3600
```

**power\_ac\_limit (v9+ only)** optional, integer, >0 and <=1000000, in watts

Custom ac power limitation imposed on the solar inverter in watts (W). Values for 'power\_ac\_limit' which exceed 'power\_ac\_max' will be rounded down to 'power\_ac\_max'.

Example:

```
3600
```

**efficiency\_max** float,  $\geq 0.5$  and  $\leq 1$

Example:

0.97

**mpp\_voltage\_min** integer,  $> 0$  and  $\leq 1000$ , in volts

Example:

175

**mpp\_voltage\_max** integer,  $> 0$  and  $\leq 1000$ , in volts

Example:

500

**mppt\_count** integer,  $> 0$  and  $\leq 100$

Example:

2

**i\_mppt\_max** float,  $\geq 0$  and  $\leq 1000$ , (**i\_mppt\_max** and **i\_dc\_max** cannot both be equal to 0)

Example:

23

**i\_dc\_max** float,  $\geq 0$  and  $\leq 1000$ , (**i\_mppt\_max** and **i\_dc\_max** cannot both be equal to 0)

Example:

0

**with\_transformer** integer, 0 or 1, 0 = without transformer, 1 = with transformer

Example:

0

**efficiency\_type** integer, one of 0500, 0505, 0510, 0515, 1000, 1005, 1010, 1015, 1020, 2000, 2005, 2010, 2015, 2020, 3000, 3005, 3010, 3015, 3020, 4000, 4005, 4010, 4015, 4020, 4030, 4040, 5000, 5010, 5020, 6000, 6010, 6020, 6030, 8040

Example:

3010

*Note: Each 4-digit value represents a different pre-defined inverter efficiency curve. The left and right pair of digits is defined as the efficiency reduction at 10% respectively 100% of 'ac\_power\_max' relative to 'efficiency\_max'. A pair of digits equals a percentage x 10. Example: A solar inverter with 98% efficiency\_max and efficiency\_type '4015' defines an inverter efficiency curve with  $98 - 4.0 = 94\%$  and  $98 - 1.5 = 96.5\%$  efficiency at 10% respectively 100% of power\_ac\_max.*

## **cable\_dc**

Both components of *cable\_dc* are mandatory.

***cable\_length*** integer, >0 and <=100000, in meters, total length without module cables

Example:

```
40
```

***cable\_thickness*** float, >=1 and <=10, thickness in mm<sup>2</sup>

Example:

```
4
```

## skyline

The skyline component allows for modelling the shading effects of solar modules. Here the skyline is a 2D-representation of the surroundings as viewed from the solar module. The actual skyline consists of an array with exactly 24 items. Each item represents the apparent height (in degrees) of obstacles blocking the skydome for the corresponding azimuth. Azimuth starts at 0 degrees (North) and increases clockwise (N-E-S-W) in steps of 15 degrees to 345 degrees, leading to the aforementioned 24 values. The values have to be  $\geq 0$  and  $\leq 90$  and must be divisible by 5 (e.g. 0, 5, 10, 15, 20, etc). Only the values in the array will be validated, therefore both examples below are valid.

Example of a 'skyline' containing a single skyline:

```
[
  [5,10,5,5,10,5,5,10,5,5,10,5,5,10,5,5,10,5,5,10,5,5,10,5]
]
```

ór

```
{
  { 0: 5, 15: 10, 30: 5, 45: 5, 60: 10, 75: 5, 90: 5, 105: 10, 120: 5, 135: 5, 150: 10,
    165: 5, 180: 5, 195: 10, 210: 5, 225: 5, 240: 10, 255: 5, 270: 5, 285: 10, 300: 5, 315:
    5, 330: 10, 345: 5 }
}
```

Multiple skylines.

Creating multiple skylines is particularly useful in case the skydome is blocked by nearby obstacles. For nearby obstacles the azimuth and/or height of the obstacle may vary for individual solar modules in the array because of parallax effects. Using multiple skylines greatly improves realistic modelling of shading effects in partly or unevenly shaded strings.

Example of a 'skyline' containing multiple skylines:

```
[
  [35,25,20,10,30,25,25,20,15,15,30,35,20, 5, 5, 5, 5, 5,55,55,45,45,45,40],
  [35,25,10,10,30,25,25,20,15,15,30,25,45,15, 5, 5, 5, 5,55,55,45,45,45,40],
  [30,20,10,10,30,25,25,20,15,15,35,25,25,50, 5, 5, 5, 5, 5,50,50,40,40,35],
  [30,20,10,30,25,30,20,20,15,15,40,35,30,20,55,55, 5, 5, 5,50,50,40,40,35],
  [25,20,10,30,25,30,20,20,15,15,45,45,35,25, 5, 5,55,50, 5,45,35,35,35,30],
  [25,20,10,30,25,30,20,20,15,15,50,50,40,30, 5, 5, 5,50,45,45,35,35,35,30]
]
```

# getSolarModules

This method retrieves a list with all the standard available SolarModules in the system.

## URL

/getSolarModules/

## Method

POST

## Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

## Success Response Example

```
[
  {
    "id": "1",
    "name": "SolarModule One",
    "efficiency_type": "045",
    "power_stc": "250",
    "voltage_mpp": "29.9",
    "current_mpp": "8.3612",
    "efficiency_stc": "0.155",
    "temp_coefficient": "0.0042",
    "absorption_coeff": "0.85",
    "reduced_efficiency_at_low_radiation": "0.045",
    "ar_coating": "1",
    "cable_length": "2",
    "cable_thickness": "4"
  }
]
```

## Error Response Example

```
{
  "error": {
    "code": 1000,
    "message": "User cannot be authenticated; invalid credentials provided"
  }
}
```

# getSolarInverters

This method returns a list with all the standard available SolarInverters in the system.

## URL

/getSolarInverters/

## Method

POST

## Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

## Success Response Example

```
[
  {
    "id": "1",
    "name": "SolarInverter One",
    "efficiency_type": "3010",
    "power_ac_max": "3600",
    "power_ac_limit": "3600",
    "efficiency_max": "0.97",
    "mpp_voltage_min": "175",
    "mpp_voltage_max": "500",
    "mppt_count": "2",
    "i_mppt_max": "23",
    "i_dc_max": "0",
    "with_transformer": "0"
  }
]
```

## Error Response Example

```
{
  "error": {
    "code": "1001",
    "message": "1.2.3.4 is not in whitelist for user myUsername"
  }
}
```

# getSolarArrayMountings

Use this method go retrieve all the possible valid values for "mounting".

## URL

/getSolarArrayMountings/

## Method

POST

## Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

## Success Response Example

```
[
  {
    "id": "1",
    "string": "on roof",
    "name": "Op dak"
  },
  {
    "id": "2",
    "string": "in roof",
    "name": "In dak"
  },
  ...etc...
]
```

## Error Response Example

```
{
  "error": {
    "code": 1002,
    "message": " Throttle limit reached "
  }
}
```

# getMeteoStations

This method can be used to get a list of all MeteoStations and their details.

## URL

/getMeteoStations/

## Method

POST

## Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

## Success Response Example

```
[
  {
    "id": "210",
    "name": "VALKENBURG",
    "latitude": "52.165",
    "longitude": "4.419",
    "altitude": "-0.2"
  },
  {
    "id": "235",
    "name": "DE KOOY",
    "latitude": "52.924",
    "longitude": "4.785",
    "altitude": "0.5"
  },
  ...etc...
]
```

## Error Response Example

```
{
  "error": {
    "code": 1003,
    "message": " You do not have sufficient credits left in your account "
  }
}
```

## getDatesLatestMeteo

This method can be used to get a list of all MeteoStation ID's and their accompanying latest date for which daily meteorological data is available.

### URL

/getDatesLatestMeteo/

### Method

POST

### Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

### Success Response Example

```
[
  {
    "id": "210",
    "date": "2013-03-25"
  },
  {
    "id": "235",
    "date": "2013-03-26"
  },
  {
    "id": "240",
    "date": "2013-03-25"
  },
  ...etc...
]
```

### Error Response Example

```
{
  "error": {
    "code": 1001,
    "message": "1.2.3.4 is not in whitelist for user myUsername"
  }
}
```

## getDatesLatestHourly

This method can be used to get a list of all MeteoStation ID's and their accompanying latest date for which hourly meteorological data is available.

### URL

/getDatesLatestHourly/

### Method

POST

### Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

### Success Response Example

```
[
  {
    "id": "210",
    "date": "2013-03-25"
  },
  {
    "id": "235",
    "date": "2013-03-26"
  },
  {
    "id": "240",
    "date": "2013-03-25"
  },
  ...etc...
]
```

### Error Response Example

```
{
  "error": {
    "code": 1001,
    "message": "1.2.3.4 is not in whitelist for user myUsername"
  }
}
```

## getMaximumNumberOfArrays

This method is used to retrieve the current maximum number of arrays that can be passed to *getCalculationResult*.

### URL

/getMaximumNumberOfArrays/

### Method

POST

### Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

### Success Response Example

```
{
  "max_arrays": "25"
}
```

### Error Response Example

```
{
  "error": {
    "code": 1005,
    "message": " Your IP has been blacklisted "
  }
}
```

# getCredits

This method returns the number of credits left in your account.

## URL

/getCredits/

## Method

POST

## Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

## Success Response Example

```
{  
  "credits": "10"  
}
```

## Error Response Example

```
{  
  "error": {  
    "code": 1000,  
    "message": "User cannot be authenticated; invalid credentials provided"  
  }  
}
```

# getThrottlingSettings

This method returns information about the current settings for rate-limiting.

## URL

/getThrottlingSettings/

## Method

POST

## Data Parameters

*login* string

Example:

```
"myUsername"
```

*password* string

Example:

```
"myPassword"
```

## Success Response Example

```
{
  "throttle_limit": "30",
  "throttle_interval": "10"
}
```

## Error Response Example

```
{
  "error": {
    "code": 1005,
    "message": "Your IP has been blacklisted"
  }
}
```

## getVersions

This method returns all possible simulator version strings for use with *getCalculationResult*.

“V9” is the stable version (C++ compiled library).

“v7” is obsolete but kept available for legacy users. DO NOT USE.

### URL

/getVersions/

### Method

POST

### Data Parameters

**login** string

Example:

```
"myUsername"
```

**password** string

Example:

```
"myPassword"
```

### Success Response Example

```
[ "v7", "v9" ]
```

### Error Response Example

```
{
  "error": {
    "code":1005,
    "message":"Your IP has been blacklisted"
  }
}
```

## Appendix A: API Errors

code	message
1000	User cannot be authenticated; invalid credentials provided
1001	<IP> is not in whitelist for user <username>
1002	Throttle limit reached
1003	You do not have sufficient credits left in your account
1004	Your account has been temporarily disabled; please contact Siderea
1005	Your IP has been blacklisted
2001	start_date missing
2002	end_date missing
2003	latitude missing
2004	longitude missing
2005	meteo_stations missing
2006	arrays missing
2007	Invalid latitude provided
2008	Invalid longitude provided
2009	No meteorological data available for provided start_date
2010	Invalid start_date provided; date should be in ISO8601 format (YYYY-mm-dd)
2011	No meteorological data available for provided end_date
2012	Invalid end_date provided; date should be in ISO8601 format (YYYY-mm-dd)
2013	Invalid input for meteo_stations provided
2014	No information for arrays provided
2015	Limit for arrays exceeded; current maximum is set at <nr>
2016	Incomplete data for Array <nr>
2017	Incomplete data for array_layout in Array <nr>
2018	Incomplete data for solar_module in Array <nr>
2019	Incomplete data for solar_inverter in Array <nr>
2020	Incomplete data for cable_dc in Array <nr>
2021	Invalid azimuth for array_layout in Array <nr>
2022	Invalid tilt for array_layout in Array <nr>
2023	Invalid mounting for array_layout in Array <nr>
2024	Invalid module_count for array_layout in Array <nr>
2025	Invalid string_count for array_layout in Array <nr>
2026	Invalid inverter_count for array_layout in Array <nr>
2027	Invalid power_stc for solar_module in Array <nr>
2028	Invalid reduced_efficiency_at_low_radiation for solar_module in Array <nr>
2029	Invalid voltage_mpp for solar_module in Array <nr>
2030	Invalid current_mpp for solar_module in Array <nr>
2031	Invalid efficiency_stc for solar_module in Array <nr>
2032	Invalid temp_coefficient for solar_module in Array <nr>
2033	Invalid cable_length for solar_module in Array <nr>
2034	Invalid power_ac_max for solar_inverter in Array <nr>
2035	Invalid with_transformer for solar_inverter in Array <nr>
2036	Invalid efficiency_max for solar_inverter in Array <nr>
2037	Invalid mpp_voltage_min for solar_inverter in Array <nr>

2038 Invalid mpp\_voltage\_max for solar\_inverter in Array <nr>  
2039 Invalid mppt\_count for solar\_inverter in Array <nr>  
2040 Invalid cable\_length for cable\_dc in Array <nr>  
2041 Invalid cable\_thickness for cable\_dc in Array <nr>  
2042 Invalid skyline in Array <nr>  
2043 Total Module Power exceeds the Total Inverter Output Power too much for Array <nr>  
2044 String Voltage exceeds certain limitations during extreme weather conditions for Array <nr>  
2045 Invalid input for arrays provided; view implementation manual for details  
2046 Invalid inverter\_shared\_with in Array <nr>  
2047 Array cannot share inverter with itself for Array <nr>  
2048 Array cannot share inverter with an array that is not sharing its inverter for Array <nr>  
2049 Provided start\_date is after end\_date  
2050 Incomplete meteorological data detected (not contiguous)  
2051 Invalid id for solar\_module in Array <nr>  
2052 Invalid id for solar\_inverter in Array <nr>  
2053 No MeteoStations available; please contact Siderea  
2054 output\_type missing  
2055 Invalid output\_type provided  
2056 There is something wrong with the permissions for your account; please contact support  
2057 An unexpected internal error occurred  
2058 Invalid key for array in arrays-parameter; keys should be consecutive, starting from 1  
2059 Invalid value for inverter\_shared\_with in Array <nr>  
2060 Inverters are not equal whereas arrays are sharing their inverter (arrays <nr> and <nr>)  
2061 Invalid efficiency\_type for solar\_inverter in Array <nr>  
2062 simulator\_version missing  
2063 Invalid simulator\_version provided  
2064 Invalid power\_ac\_limit for solar\_inverter in Array <nr>  
2065 Invalid absorption\_coeff for solar\_module in Array <nr>  
2066 Invalid ar\_coating for solar\_module in Array <nr>  
2067 Invalid cable\_thickness for solar\_module in Array <nr>  
2068 Invalid i\_mppt\_max for solar\_inverter in Array <nr>  
2069 Invalid i\_dc\_max for solar\_inverter in Array <nr>  
2070 Invalid combination of i\_mppt\_max and i\_dc\_max for solar\_inverter in Array <nr>  
2071 Invalid i\_dc\_max for solar\_inverter in Array <nr>  
2072 code\_sq missing  
2073 Invalid code\_sq provided  
2074 Invalid meteorological data provided (or invalid julian day)  
2075 Invalid value for powerlimiter at Array <nr>  
2076 Shared inverter is connected with more than one powerlimiter <nr>  
2077 More than one powerlimiter in use in Array <nr>  
2078 Insufficient inputpower on powerlimiter to reduce total outputpower  
2079 Number of strings cannot exceed number of modules  
2080 Invalid efficiency\_type for solar\_module in Array <nr>