Core API v1

What is the Eniscope Core API?

The Eniscope Core API is a RESTful HTTP API that provides access to stored metering data along with metadata connected to the metering data organized in an Organization hierarchy. Access to the data stored on the Eniscope Platform is restricted to where in the hierarchy your account sits.

This documentation is aimed at getting a developer up to speed with requesting metering data from the Eniscope Platform. It provides everything needed to find devices and data channels from the available organizations and request metering data from those devices.

Accessing the API & Authentication

Accessing the API

The Live API can be accessed via `https://core.eniscope.com/`

Authentication

To access the Eniscope API, the application that uses the API must provide authentication using both Basic Authentication and an API key. Contact the support team to request an API key, they will configure and add a new Service to the Organization requiring API access.

Every request needs to include a valid Eniscope Platform user email address and the md5 hash of the user password or a user session token. The token is returned in the HTTP headers after a successful user login attempt.

To authenticate with the Eniscope API, follow these steps:

- Obtain the user's email address, this will be used as the username.
  e.g. `api@someorg.com`

- Calculate the MD5 hash of the user's password.
  e.g. for the plain text password `thisismypassword` the MD5 hash would be `31435008693ce6976f45dedc5532e2c1`

- Create the Basic Authentication header that includes the username and the calculated MD5 hash as the password. The following is an example of how to include Basic Authentication in your request header:
  `Authorization: Basic {Base64Encode(emailAddress:MD5Password)}`
  Replace `{Base64Encode(emailAddress:MD5Password)}` with the Base64-encoded combination of the user's email address and MD5 password hash.
  e.g. using the example email address and password above
  `{Base64Encode(api@someorg.com:31435008693ce6976f45dedc5532e2c1)}` the Base64 encode would be
  `YXBpQHNvbWVvcmcuY29tOjMxNDM1MDA4NjkzY2U2OTc2ZjQ1ZGVkYzU1MzJlMmMx`

- Include the API key in the request header with the index `X-Eniscope-API`.
  `X-Eniscope-API: {APIKey}`
  Replace `{APIKey}` with the actual API key which was provided by the support team.

- Once the user login has been verified, response headers will include a token, by including this token header in all subsequent requests, no user login details are required while the session is still valid.
  `X-Eniscope-Token: {SessionToken}`

Example cURL call

```
1  curl --location --request GET 'https://core.eniscope.com/devices' \
2  --header 'X-Eniscope-API: APIKey' \
3  --header 'Authorization: Basic YXBpQHNvbWVvcmcuY29tOjMxNDM1MDA4NjkzY2U2OTc2ZjQ1ZGVkYzU1MzJlMmMx'
```

Security Recommendations

- The API key should be kept confidential and should not be shared publicly.
- Use HTTPS to encrypt data transmitted between your application and the Eniscope API to ensure secure communication.

Access Control

Access to specific resources and endpoints may be restricted based on user roles and permissions. Ensure that you have the necessary access rights to perform the desired operations.

Collections & Relational data

Collections are subsets of resources associated with a specific parent resource. They allow related data to be retrieved that belong to a particular resource. Collections can be added to a resource's endpoint to retrieve additional information or related records.

Collections provide a convenient way to access related data without making separate API calls for each piece of information.

The availability of collections may vary based on the resource type and its relationships.

Usage

To access a collection associated with a resource, append the collection name to the resource's endpoint URL. For example:

`/resource/{resourceId}/collection` - Retrieves a collection associated with the specified resource.

Multiple collections can be included in a single request

Examples

- Retrieve Alarm Rules for a Specific Alarm

  To retrieve alarm rules linked to a specific alarm, use the following endpoint:

  `GET /alarms/{alarmId}/alarmrules`

  This call will return all the alarm rules associated with the specified alarm.

- Retrieve Events for a Device

  To retrieve events related to a specific device, you can use the following endpoint:

  `GET /devices/{deviceId}/events`

  This call will return all events associated with the specified device.

- To retrieve the list of meters and tariffs for a device

  `GET /devices/{deviceId}/meters/tariffs/`

Response

Within the API, requested collections are efficiently embedded within the JSON payload of the parent resource. These collections are presented as arrays, neatly organized and accessible by their specific collection names. This design facilitates seamless navigation and interaction with the associated collections, all conveniently encapsulated within the response of the parent resource. An example response is as follows:

```
1  {
2      "accountingCode" : null,
3      "address1" : "Some Organization",
4      "address2" : "",
5      "address3" : "",
6      "addressId" : "17322",
7      "appliableSubscriptionId" : null,
8      "carbonRatio" : "0.0000",
9      "carbonUnits" : "",
10     "city" : "London",
11     "country" : "United Kingdom",
12     "createdBy" : "14491",
13     "createdTs" : "1659361785",
```

```
14        "currencyCode" : "GBP",
15        "currencySymbol" : "£",
16        "defaultAddressId" : "7322",
17        "defaultEmailAddress" : "api@someorg.com",
18        "defaultLanguageCode" : "en",
19        "distributor" : {},
20        "divisionLabel" : "",
21        "fax" : null,
22        "groupId" : null,
23        "lastLogIn" : null,
24        "lat" : null,
25        "licenseDiscount" : null,
26        "links" : {
27          "accounts" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/accounts",
28          "addresses" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/addresses",
29          "ancestry" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/ancestry",
30          "channels" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/channels",
31          "clientissues" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/clientissues",
32          "devices" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/devices",
33          "events" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/events",
34          "expired" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/expired",
35          "issues" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/issues",
36          "issuesold" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/issuesold",
37          "leaderboard" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/leaderboard",
38          "meters" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/meters",
39          "metersbelow" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/metersbelow",
40          "pagedmeters" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/pagedmeters",
41          "pagedmetersbelow" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/pagedmetersbelow",
42          "readings" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/readings",
43          "self" : "https://core-lb.prod.best.energy/v1/1/organizations/12345",
44          "services" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/services",
45          "stats" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/stats",
46          "tags" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/tags",
47          "tariffs" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/tariffs",
48          "timeperiods" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/timeperiods",
49          "trendlines" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/trendlines",
50          "webhooks" : "https://core-lb.prod.best.energy/v1/1/organizations/12345/webhooks"
51        },
52        "lng" : null,
53        "logInCounter" : null,
54        "logo" : null,
55        "marketingOptIn" : "0",
56        "meters" : [
57          {
58            "binaryStoreId" : "2",
59            "channels" : "3",
60            "createdBy" : "14491",
61            "createdTs" : "1659361993",
62            "ct" : "3000",
63            "dataPlan" : "12",
64            "dataReceivedTs" : null,
65            "dataType" : "grid meter",
66            "deviceId" : "68023",
67            "deviceName" : "Meter 1",
68            "deviceTypeId" : "20",
69            "deviceTypeInterfaceId" : "25",
70            "expires" : "1732492800",
71            "interface" : {
```

```json
 72            "count" : "8",
 73            "createdBy" : "1",
 74            "createdTs" : "1521800190",
 75            "deviceInterfaceId" : "3",
 76            "deviceTypeId" : "16",
 77            "deviceTypeInterfaceId" : "25",
 78            "direction" : "accept",
 79            "interfaceName" : "Onboard",
 80            "links" : {
 81                "self" : "https://core-lb.prod.best.energy/v1/1/deviceTypeInterfaces/25"
 82            },
 83            "modifiedBy" : null,
 84            "modifiedTs" : null,
 85            "prefix" : "00"
 86          },
 87          "isLeadingTrailing" : null,
 88          "links" : {
 89              "self" : "https://core-lb.prod.best.energy/v1/1/meters"
 90          },
 91          "meterId" : "52454",
 92          "modifiedBy" : "14491",
 93          "modifiedTs" : "1659367449",
 94          "nominalCode" : null,
 95          "organizationId" : "12345",
 96          "parentId" : "68022",
 97          "pulseEnergyRatio" : null,
 98          "pulseLabel" : "",
 99          "pulseRatio" : null,
100          "pulseUnits" : "",
101          "registered" : "1659361993",
102          "renewed" : null,
103          "serviceAgreementId" : null,
104          "sortCode" : "1-2-23456-12345-",
105          "status" : "1",
106          "subscriptionId" : null,
107          "temperatureEnabled" : "0",
108          "txm" : null,
109          "uuId" : "6A2719BDAEA50001",
110          "vendor" : null,
111          "vendor_format_id" : null,
112          "vendor_id" : null,
113          "vendor_parent_id" : null,
114          "vs" : "415"
115        },
116      ],
117      "modifiedBy" : "13761",
118      "modifiedTs" : "1659546659",
119      "nominalCode" : null,
120      "organizationDocumentsUrl" : null,
121      "organizationId" : "12345",
122      "organizationKey" : null,
123      "organizationName" : "Some Organization",
124      "organizationType" : null,
125      "parentId" : "23456",
126      "postCode" : "",
127      "regId" : null,
128      "sortCode" : "1-2-23456-12345-",
129      "state" : "",
```

```
130     "status" : "1",
131     "taxCode" : null,
132     "telephone" : null,
133     "temperatureType" : "Celsius",
134     "thirdPartyOptIn" : "0",
135     "timeZone" : "Etc/GMT"
```

Filters, Sorting & Pagination

Response Content Type

By including the Accept Header, the format of the returning resource data can be set. The API can accept either `Accept:text/json` or `Accept:text/xml`

> ℹ The default response type will be '`text/json`'.

Request Query Filters

There are a number of ways to filter a list of resources returned by a resource end-point. Each resource has a allowed list of filters that can be used to refine the list of resources returned.

The list of filters can be found for each resource by using the HTTP OPTIONS Request

    OPTIONS /devices/ Accept: text/json

The filters can then be used to filter the list of resources returned.

- For example, the following will return all devices with `deviceType` equal to 6

    GET /devices/?deviceType=6

The filters can be combined with wildcards to return the desired list.

- **Like**: `?deviceName=*Eniscope`
- **Equal to**: `?deviceType=6`
- **Not Equal to**: `?deviceType=!6`
- **Not Null**: `?deviceType=!null`
- **Null**: `?deviceType=null`

Multiple filters can appear more than once to allow for more complex queries

- **Like OR Null**: `?deviceName=*Eniscope*&deviceName=null`

Sorting

Filters can also be used as sorts on the returned list of resources, for example:

- **Sort**: `?sort=deviceName`
- **Sort Ascending**: `?sort=deviceName asc`
- **Sort Descending**: `?sort=deviceName desc`

Pagination

When requesting lists of resources responses are split into pages with each page defaulting to 20 records per page.

To go to another page, add `?page={page number}` onto the request query string. The data in the resource list metadata property provides information about the number of records, pages and the number of records per page.

    GET /devices/?page=6

It is possible to change the number of records returned per page by including the '`limit`' option. The maximum allowed number of records per page is 100.

    GET /devices/?page=6&limit=50

Status Reponse Codes

Statuses from the API are provided as HTTP response statuses including any exceptions thrown by the API.

The following are the main status codes used by the Core API

Status Code	Description	
200	OK	Success status response code indicates that the request has succeeded
401 or 419	Unauthorised	The response status code indicates that the client request has not been completed because it lacks valid authentication credentials for the requested resource.
402	Payment Required	The subscription has expired
403	Forbidden	Invalid credentials. The response status code indicates that there are insufficient rights to access the resource.
412	Precondition Failed	Prerequisite not met and access to the target resource has been denied.
404	Not Found	The response status code indicates that the server cannot find the requested resource.
409	Conflict	The response status code indicates a request conflict or a duplicate of the target resource.
500	Internal Server Error	A server error response code indicates that the server encountered an unexpected condition that prevented it from fulfilling the request. This error response is a generic "catch-all" response.

Core API Endpoints

Data in the Core API is organised in an Organization hierarchy with devices being added to the organization where they have been installed or against the organization that owns the devices.

Due to the complex setup of some metering point types with multiple options on how you would want to view the data. The viewing of meter readings has been split into another endpoint called 'channels'. In order to view the readings for a metering point you need to use one of the channels that have been set up for that device, it is worth noting that you can have several channels per metering point with each channel viewing the data for that metering point in a different way. For example, one channel might be set up to only display 'phase 1' from that metering point and another might be only to display the 'phase' counter attached to that metering point.

GET /accounts

Retrieve a list of accounts

`GET` `GET /accounts`

Parameters

`name`	Optional	Filter by account name.
`email`	Optional	Filter by email address.
`group`	Optional	Filter by group ID.
`status`	Optional	Filter by account status.
`organization`	Optional	Filter by organization ID.
`search`	Optional	Search for accounts by email address or account name

Example Request

```
1  GET /accounts?name=John&email=john@example.com&group=123&status=active&organization=456&search=example
```

Retrieve a specific account by ID

`GET` `GET /accounts/{accountId}`

Parameters

`accountId`	mandatory	The account ID.

Example Request

```
1  GET /accounts/{accountId}
```

GET Response Body

The response is in JSON format and includes the following fields:

Field	Data Type	Description
`accountId`	int	The unique identifier for the account.
`organizationId`	int	The identifier of the organization associated with the account.
`groupId`	int	The identifier of the group to which the account belongs.

parentId	int or null	The identifier of the parent account, if applicable.
nominalCode	string or null	The nominal code associated with the account.
accountName	string	The name of the account holder.
emailAddress	string	The email address associated with the account.
status	int	The account status (e.g., "1" for active).
extendedLogin	int or null	Extended login session setting.
logInCounter	int	The login counter for the account.
lastLogIn	string	The timestamp of the last login.
resetCode	string or null	Reset code information, if available.
createdBy	int	The identifier of the user who created the account.
modifiedBy	int or null	The identifier of the user who last modified the account.
createdTs	string	The timestamp of when the account was created.
modifiedTs	string or null	The timestamp of when the account was last modified.
mfaToken	string or null	Multi-factor authentication token, if enabled.
mfaTokenSentTs	string or null	Timestamp of when the MFA token was sent.
mfaSuccessfulTs	string or null	Timestamp of a successful MFA attempt.
failedLoginCount	string or null	Count of failed login attempts.
loginLockedTs	string or null	Timestamp when the login was locked, if applicable.
passwordModifiedTs	string	Timestamp of the last password modification.
datastorePreference	string or null	Datastore preference, if available.
sortCode	string	Sort code associated with the account.

```
1   {
2       "accountId": "1",
3       "links": {
4           "self": "https://core.eniscope.com/v1/1/accounts/1",
5           "devices": "https://core.eniscope.com/v1/1/accounts/1/devices",
6           "channels": "https://core.eniscope.com/v1/1/accounts/1/channels",
7           "tags": "https://core.eniscope.com/v1/1/accounts/1/tags",
8           "organizations": "https://core.eniscope.com/v1/1/accounts/1/organizations",
9           "resources": "https://core.eniscope.com/v1/1/accounts/1/resources",
10          "events": "https://core.eniscope.com/v1/1/accounts/1/events"
11      },
12      "organizationId": "1",
13      "groupId": "1",
14      "parentId": null,
```

```json
15      "nominalCode": null,
16      "accountName": "John Doe",
17      "emailAddress": "john.doe@example.com",
18      "status": "1",
19      "extendedLogin": null,
20      "logInCounter": "23025",
21      "lastLogIn": "1494560701",
22      "resetCode": null,
23      "createdBy": "15552",
24      "modifiedBy": null,
25      "createdTs": "1685535742",
26      "modifiedTs": null,
27      "mfaToken": null,
28      "mfaTokenSentTs": null,
29      "mfaSuccessfulTs": null,
30      "failedLoginCount": null,
31      "loginLockedTs": null,
32      "passwordModifiedTs": "1654767213",
33      "datastorePreference": null,
34      "sortCode": "1-"
35  }
```

GET /alarmperiods

Retrieve a list of alarm periods

`GET` `GET /alarmperiods`

Parameters

`alarm`	Optional	Filter by alarm ID.

Example Request

```
1  GET /alarmperiods?alarm=123
```

Retrieve a specific alarm period by ID

`GET` `GET /alarmperiods/{periodId}`

Parameters

`periodId`	Optional	Period ID.

Example Request

```
1  GET /alarmperiods/{periodId}
```

GET /alarmrules

Retrieve a list of alarm rules

`GET` `GET /alarmrules`

Parameters

alarm	Optional	Filter by alarm ID.

Example Request

```
1  GET /alarmrules?alarm=123
```

Retrieve a specific alarm rule by ID

`GET` `GET /alarmrules/{ruleId}`

Parameters

ruleId	Mandatory	Rule ID

Example Request:

```
1  GET /alarmrules/{ruleId}
```

GET /alarms

Retrieve a list of alarms

`GET` `GET /alarms`

Parameters

`name`	Optional	Filter by account name.
`organization`	Optional	Filter by organization ID.
`uuId`	Optional	Filter by device UUID.
`channel`	Optional	Filter by channel ID.
`type`	Optional	Filter by alarm type.
`displayedPhase`	Optional	Filter by displayed phase.
`status`	Optional	Filter by alarm status.

Example Request:

```
1  GET /alarms?organization=123&channel=456&type=high&status=active
```

Retrieve a specific alarm by ID

`GET` `GET /alarms/{alarmId}`

Parameters

`alarmId`	Mandatory	Alarm ID.

Example Request

```
1  GET /alarms/{alarmId}
```

GET Response Body

The response is in JSON format and includes the following fields:

Field	Data Type	Description

`alarmId`	int	Unique identifier for the alarm.
`alarmName`	string	Name of the alarm.
`alarmDescription n`	string	Description of the alarm.
`channelId`	int	Identifier of the associated channel.
`deviceId`	int	Identifier of the associated device.
`organizationId`	int	Identifier of the associated organization.
`emailRecipient s`	string	Email recipients for the alarm notifications.
`emailTemplateI d`	int	Identifier of the email template, if applicable.
`emailLanguage`	string	Language for email notifications.
`alarmInterval`	int	Interval for the alarm.
`reportingInterv al`	int	Reporting interval for the alarm.
`reminderInterva l`	int	Reminder interval for the alarm.
`alarmHoldoffCou nt`	int	Count of alarm holdoffs.
`status`	int	Alarm status (e.g., "1" for active).
`createdBy`	int	Identifier of the user who created the alarm.
`modifiedBy`	int	Identifier of the user who last modified the alarm.
`createdTs`	string	Timestamp when the alarm was created.
`modifiedTs`	string	Timestamp when the alarm was last modified.
`uuId`	string	UUID of the alarm.
`timeZone`	string	Time zone associated with the alarm.
`expires`	string	Timestamp when the alarm expires.
`displayedPhase`	string	Displayed phase for the alarm.
`pulseRatio`	int	Pulse ratio for the alarm.
`pulseLabel`	string	Pulse label for the alarm.
`links.self`	string	URL for accessing this alarm resource.
`links.alarmrule s`	string	URL for accessing alarm rules related to this alarm.
`links.alarmperi ods`	string	URL for accessing alarm periods related to this alarm.

```json
{
    "alarmId": "75",
    "alarmName": "Office Energy Usage",
    "alarmDescription": "This alarm checks the energy usage of the office",
    "channelId": "772",
    "deviceId": "963",
    "organizationId": "10039",
    "emailRecipients": "spoofed-email@eniscope.com, spoofed-email2@eniscope.com",
    "emailTemplateId": null,
    "emailLanguage": "es",
    "alarmInterval": null,
    "reportingInterval": "900",
    "reminderInterval": "900",
    "alarmHoldoffCount": "1",
    "status": "1",
    "createdBy": "1469439186",
    "modifiedBy": "1",
    "createdTs": "1",
    "modifiedTs": "1469439295",
    "uuId": "001BE510145F0000",
    "timeZone": "Europe/London",
    "expires": "1572566400",
    "displayedPhase": "system",
    "pulseRatio": null,
    "pulseLabel": null,
    "links": {
        "self": "https://core.eniscope.com/v1/1/alarms/75",
        "alarmrules": "https://core.eniscope.com/v1/1/alarms/75/alarmrules",
        "alarmperiods": "https://core.eniscope.com/v1/1/alarms/75/alarmperiods"
    }
}
```

GET /channels

A channel in the Eniscope Platform is a configuration record that controls how data from a device metering point is viewed and accessed.

This enables a 3-phase metering point to be viewed as 3 separate channels each with its own description and tariffs. It also enables pulse values to be viewed independently from its metering point.

> ℹ️ Metering Data Can only be accessed via a Data Channel

Retrieve a list of channels

`GET` `GET /channels`

Parameters

`displayedPhase`	Optional	Filter by displayed phase.
`name`	Optional	Filter by channel name.
`organization`	Optional	Filter by organization ID.
`tariff`	Optional	Filter by tariff ID.
`type`	Optional	Filter by device type ID.
`uuid`	Optional	Filter by device UUID.
`tag`	Optional	Filter by tag ID.
`sortOrder`	Optional	Filter by sort order.
`deviceTypeId`	Optional	Filter by device type ID.
`deviceId`	Optional	Filter by device ID.
`device`	Optional	Filter by device ID.

Example Request

```
1  GET /channels?organization=123&name=example&tag=456
```

Retrieve a specific channel by ID

`GET` `GET /channels/{channelId}`

Parameters

`channelId`	Mandatory	Channel ID.

Example Request

```
1   GET /channels/{channelId}
```

GET /customparameters

Retrieve a list of custom parameters

`GET` `GET /customparameters`

Parameters

`organizationId`	Optional	Filter by organization ID.
`parameterName`	Optional	Filter by parameter name.
`parameterLabel`	Optional	Filter by parameter label.

GET Response Body

The response is in JSON format and includes the following fields:

Field	Data Type	Description
`id`	int	The unique identifier for the custom parameter.
`organizationId`	int	The identifier of the organization associated with the custom parameter.
`parameterLabel`	string	The human-readable name of the custom parameter.
`parameterName`	string	The name of the custom parameter.
`parameterUnits`	string or null	The units used with readings of the custom parameter (if any).
`valueType`	string	Either 'instant' or 'counter'.
`dataType`	string	The valid CrateDB data type that is used for this custom parameter.
`createdBy`	int	The identifier of the user who created the account.
`createdTs`	string	The timestamp of when the account was created.

```
1   {
2       "result": [
3           {
4               "id": "1",
5               "organizationId": "1",
6               "parameterLabel": "new param name 1",
7               "parameterName": "npn1",
8               "parameterUnits": "psi",
9               "valueType": "instant",
10              "dataType": "DOUBLE PRECISION",
```

```
11              "createdBy": "13904",
12              "createdTs": "1713246845"
13          },
14          {
15              "id": "2",
16              "organizationId": "1",
17              "parameterLabel": "new param name 2",
18              "parameterName": "npn2",
19              "parameterUnits": "psi",
20              "valueType": "instant",
21              "dataType": "DOUBLE PRECISION",
22              "createdBy": "13904",
23              "createdTs": "1713247261"
24          },
25          {
26              "id": "3",
27              "organizationId": "1",
28              "parameterLabel": "new param name 3",
29              "parameterName": "npn3",
30              "parameterUnits": "psi",
31              "valueType": "instant",
32              "dataType": "DOUBLE PRECISION",
33              "createdBy": "13904",
34              "createdTs": "1713247269"
35          },
36          {
37              "id": "4",
38              "organizationId": "1",
39              "parameterLabel": "new param name 4",
40              "parameterName": "npn4",
41              "parameterUnits": "psi",
42              "valueType": "instant",
43              "dataType": "DOUBLE PRECISION",
44              "createdBy": "13904",
45              "createdTs": "1713247275"
46          },
47          {
48              "id": "5",
49              "organizationId": "1",
50              "parameterLabel": "new param name 5",
51              "parameterName": "npn5",
52              "parameterUnits": "psi",
53              "valueType": "instant",
54              "dataType": "DOUBLE PRECISION",
55              "createdBy": "13904",
56              "createdTs": "1713247281"
57          },
58          {
59              "id": "6",
60              "organizationId": "1",
61              "parameterLabel": "new param name 6",
62              "parameterName": "npn6",
63              "parameterUnits": "psi",
64              "valueType": "instant",
65              "dataType": "DOUBLE PRECISION",
66              "createdBy": "13904",
67              "createdTs": "1713247288"
68          },
```

```
 69          {
 70              "id": "7",
 71              "organizationId": "1",
 72              "parameterLabel": "new param name 7",
 73              "parameterName": "npn7",
 74              "parameterUnits": "psi",
 75              "valueType": "instant",
 76              "dataType": "DOUBLE PRECISION",
 77              "createdBy": "13904",
 78              "createdTs": "1713247294"
 79          },
 80          {
 81              "id": "8",
 82              "organizationId": "1",
 83              "parameterLabel": "new param name 8",
 84              "parameterName": "npn8",
 85              "parameterUnits": "psi",
 86              "valueType": "instant",
 87              "dataType": "DOUBLE PRECISION",
 88              "createdBy": "13904",
 89              "createdTs": "1713247300"
 90          },
 91          {
 92              "id": "9",
 93              "organizationId": "1",
 94              "parameterLabel": "new param name 9",
 95              "parameterName": "npn9",
 96              "parameterUnits": "psi",
 97              "valueType": "instant",
 98              "dataType": "DOUBLE PRECISION",
 99              "createdBy": "13904",
100              "createdTs": "1713248112"
101          },
102          {
103              "id": "10",
104              "organizationId": "1",
105              "parameterLabel": "new param name 10",
106              "parameterName": "npn10",
107              "parameterUnits": "psi",
108              "valueType": "instant",
109              "dataType": "DOUBLE PRECISION",
110              "createdBy": "13904",
111              "createdTs": "1713248136"
112          }
113      ],
114      "meta": {
115          "currentPage": 1,
116          "recordsPerPage": 20,
117          "recordCount": 10,
118          "pageCount": 1
119      }
120  }
```

Retrieve a specific custom parameter by its ID

`GET` `GET /meters/{customParameterId}`

Parameters

`customParameterId`	Required	The unique identifier of the custom parameter.

POST+PUT /dataingress/generic

Rate Limits

POST and PUT calls are rate limited based on the number of messages currently stored in the relevant Rabbit Message Queue (RMQ), and will return a 429 when a limit has been reached. This is to avoid crashing RMQ due to many large unprocessed generic ingress messages in RMQ. As of 13th Feb 2023 the production limit is set to 10000 messages via `rmqDataIngressGenericMaxQueueSize` in application/config.ini. The production config.ini file is located here.

Insert new data

`POST` `POST /dataingress/generic`

> ℹ The POST request is limited to inserting up to 1000 rows at a time.

Update existing data (indexed by ts+uuId)

`PUT` `PUT /dataingress/generic`

> ℹ The PUT request is limited to updating up to 100 rows at a time.

> ⚠ PUT will **NOT** insert new rows.
>
> If you have a mixture of new data and updated data, you can't use PUT on it all. The new data will not get inserted without using POST.

General Parameters

API Field	Label	Type	Unit	Data Type	Notes
ts	Epoch timestamp	counter	seconds	INTEGER	
uuId	BEST Meter Custom Unique Identifier	N/A	N/A	STRING	
period	No of seconds between single (unified) readings	instant	seconds	INTEGER	Minimum = 60 Maximum = 86400 Ignored for PATCH operations
samples	No of unified samples that define this single reading	instant	samples	INTEGER	Optional
AE	Apparent Energy	counter	VAh	DOUBLE PRECISION	
AE1	Apparent Energy 1	counter	VAh	DOUBLE PRECISION	
AE2	Apparent Energy 2	counter	VAh	DOUBLE PRECISION	

27

AE3	Apparent Energy 3	counter	VAh	DOUBLE PRECISION	
BTU	BTU Absolute	counter		REAL	
C	Count	counter		REAL	
Ct	Time Since Last Pulse	counter	s	REAL	
DI1	Digital Input 1	instant		DOUBLE PRECISION	
DIt1	Time Since Last Input Event 1	instant	s	DOUBLE PRECISION	
DIC1	Digital Input Counter 1	counter		DOUBLE PRECISION	
DI2	Digital Input 2	instant		DOUBLE PRECISION	
DIt2	Time Since Last Input Event 2	instant	s	DOUBLE PRECISION	
DIC2	Digital Input Counter 2	counter		DOUBLE PRECISION	
DI3	Digital Input 3	instant		DOUBLE PRECISION	
DIt3	Time Since Last Input Event 3	instant	s	DOUBLE PRECISION	
DIC3	Digital Input Counter 3	counter		DOUBLE PRECISION	
DI4	Digital Input 4	instant		DOUBLE PRECISION	
DIt4	Time Since Last Input Event 4	instant	s	DOUBLE PRECISION	
DIC4	Digital Input Counter 4	counter		DOUBLE PRECISION	
DI5	Digital Input 5	instant		REAL	
DIt5	Time Since Last Input Event 5	instant	s	REAL	
DIC5	Digital Input Counter 5	counter		REAL	
DI6	Digital Input 6	instant		REAL	
DIt6	Time Since Last Input Event 6	instant	s	REAL	
DIC6	Digital Input Counter 6	counter		REAL	
DI7	Digital Input 7	instant		REAL	
DIt7	Time Since Last Input Event 7	instant	s	REAL	

DIC7	Digital Input Counter 7	counter		REAL	
DI8	Digital Input 8	instant		REAL	
DIt8	Time Since Last Input Event 8	instant	s	REAL	
DIC8	Digital Input Counter 8	counter		REAL	
DO1	Digital Output 1	instant		DOUBLE PRECISION	
DOt1	Time Since Last Output Event 1	instant		DOUBLE PRECISION	
DO2	Digital Output 2	instant		DOUBLE PRECISION	
DOt2	Time Since Last Output Event 2	instant		DOUBLE PRECISION	
DO3	Digital Output 3	instant		DOUBLE PRECISION	
DOt3	Time Since Last Output Event 3	instant		DOUBLE PRECISION	
DO4	Digital Output 4	instant		REAL	
DOt4	Time Since Last Output Event 4	instant		REAL	
DO5	Digital Output 5	instant		REAL	
DOt5	Time Since Last Output Event 5	instant		REAL	
DO6	Digital Output 6	instant		REAL	
DOt6	Time Since Last Output Event 6	instant		REAL	
DO7	Digital Output 7	instant		REAL	
DOt7	Time Since Last Output Event 7	instant		REAL	
DO8	Digital Output 8	instant		REAL	
DOt8	Time Since Last Output Event 8	instant		REAL	
E	Energy	counter	Wh	DOUBLE PRECISION	
E1	Energy 1	counter	Wh	DOUBLE PRECISION	
E2	Energy 2	counter	Wh	REAL	
E3	Energy 3	counter	Wh	REAL	
Ex	Export Energy	counter	Wh	REAL	
Ex1	Export Energy 1	counter	Wh	REAL	
Ex2	Export Energy 2	counter	Wh	REAL	

Ex3	Export Energy 3	counter	Wh	REAL	
M	Mass	counter	g	REAL	
PEC	People Count	counter		REAL	
PIR	PIR Count Value	counter		DOUBLE PRECISION	
PIRt	Time Since Last PIR Count	instant		DOUBLE PRECISION	
RE	Reactive Energy	counter	VArh	DOUBLE PRECISION	
RE1	Reactive Energy 1	counter	VArh	REAL	
RE2	Reactive Energy 2	counter	VArh	REAL	
RE3	Reactive Energy 3	counter	VArh	REAL	
REx	Reactive Energy Export	counter	VArh	REAL	
REx1	Reactive Energy Export 1	counter	VArh	REAL	
REx2	Reactive Energy Export 2	counter	VArh	REAL	
REx3	Reactive Energy Export 3	counter	VArh	REAL	
Ton	On Time	counter	h	BIGINT	
Top	Operating Time	counter	h	BIGINT	
Toff	Off Time	counter	h	BIGINT	
Trun	Run Time	counter	h	BIGINT	
Vol	Volume	counter	m^3	DOUBLE PRECISION	
A12	Phase Angle V1 to V2	instant	degrees	REAL	
A13	Phase Angle V1 to V3	instant	degrees	REAL	
AC	Acceleration	instant	m/s	REAL	
AEr	Apparent Energy Relative	instant	VAh	REAL	
AEr1	Apparent Energy 1 Relative	instant	VAh	REAL	
AEr2	Apparent Energy 2 Relative	instant	VAh	REAL	
AEr3	Apparent Energy 3 Relative	instant	VAh	REAL	
AF	Air Flow	instant	m/s	REAL	
AIN1	Analogue Input 1	instant	m/s	REAL	
AIN2	Analogue Input 2	instant	m/s	REAL	
AIN3	Analogue Input 3	instant	m/s	REAL	
AIN4	Analogue Input 4	instant	m/s	REAL	

AIN5	Analogue Input 5	instant	m/s	REAL	
AIN6	Analogue Input 6	instant	m/s	REAL	
AIN7	Analogue Input 7	instant	m/s	REAL	
AIN8	Analogue Input 8	instant	m/s	REAL	
AO1	Analogue Output 1	instant	m/s	REAL	
AO2	Analogue Output 2	instant	m/s	REAL	
AO3	Analogue Output 3	instant	m/s	REAL	
AO4	Analogue Output 4	instant	m/s	REAL	
AO5	Analogue Output 5	instant	m/s	REAL	
AO6	Analogue Output 6	instant	m/s	REAL	
AO7	Analogue Output 7	instant	m/s	REAL	
AO8	Analogue Output 8	instant	m/s	REAL	
AP	Atmospheric Pressure	instant	mBar	REAL	
BTUr	BTU Relative	instant		REAL	
CO	Carbon Monoxide	instant	ppm	REAL	
CO2	Carbon Dioxide	instant	ppm	REAL	
Cr	Pulse Count Relative	instant		REAL	
D	Current THD	instant	%	REAL	
D1	Current THD 1	instant	%	REAL	
D2	Current THD 2	instant	%	REAL	
D3	Current THD 3	instant	%	REAL	
Dmax	Current THD Max	instant	%	REAL	
Dmax1	Current THD 1 Max	instant	%	REAL	
Dmax2	Current THD 2 Max	instant	%	REAL	
Dmax3	Current THD 3 Max	instant	%	REAL	
DICr1	Digital Input Counter 1 relative	instant		REAL	
DICr2	Digital Input Counter 2 relative	instant		REAL	
DICr3	Digital Input Counter 3 relative	instant		REAL	
DICr4	Digital Input Counter 4 relative	instant		REAL	
DICr5	Digital Input Counter 5 relative	instant		REAL	
DICr6	Digital Input Counter 6 relative	instant		REAL	
DICr7	Digital Input Counter 7 relative	instant		REAL	

DICr8	Digital Input Counter 8 relative	instant		REAL	
Er	Energy Relative	instant	Wh	REAL	
Er1	Energy 1 Relative	instant	Wh	REAL	
Er2	Energy 2 Relative	instant	Wh	REAL	
Er3	Energy 3 Relative	instant	Wh	REAL	
Exr	Export Energy Relative	instant	Wh	REAL	
Exr1	Export Energy 1 Relative	instant	Wh	REAL	
Exr2	Export Energy 2 Relative	instant	Wh	REAL	
Exr3	Export Energy 3 Relative	instant	Wh	REAL	
F	Frequency	instant	Hz	REAL	
I	Current	instant	A	REAL	
I1	Current 1	instant	A	REAL	
I2	Current 2	instant	A	REAL	
I3	Current 3	instant	A	REAL	
Imax	Current Max	instant	A	REAL	
Imax1	Current 1 Max	instant	A	REAL	
Imax2	Current 2 Max	instant	A	REAL	
Imax3	Current 3 Max	instant	A	REAL	
In	Neutral Current	instant	A	DOUBLE PRECISION	
Inmax	Neutral Current Max	instant	A	DOUBLE PRECISION	
LL	Light Level	instant	lux	DOUBLE PRECISION	
Mf	Mass Flow	instant	g/h	REAL	
Mfr	Mass Flow Relative	instant	g/h	REAL	
NOX1	Nitrous Oxide 1	instant	ppm	REAL	
NOX2	Nitrous Oxide 2	instant	ppm	REAL	
P	Power	instant	W	REAL	
P1	Power 1	instant	W	REAL	
P2	Power 2	instant	W	REAL	
P3	Power 3	instant	W	REAL	
Pmax	Power Max	instant	W	REAL	

Pmax1	Power 1 Max	instant	W	REAL	
Pmax2	Power 2 Max	instant	W	REAL	
Pmax3	Power 3 Max	instant	W	REAL	
PE	Percentage	instant	%	REAL	
PECR	People Count Relative	instant		REAL	
PF	Power Factor	instant		REAL	
PF1	Power Factor 1	instant		REAL	
PF2	Power Factor 2	instant		REAL	
PF3	Power Factor 3	instant		REAL	
PFmax	Power Factor Max	instant		REAL	
PFmax1	Power Factor 1 Max	instant		REAL	
PFmax2	Power Factor 2 Max	instant		REAL	
PFmax3	Power Factor 3 Max	instant		REAL	
PIRr	PIR Count Value Relative	instant		REAL	
PM1	Particulate Matter 1	instant	ug/m^3	REAL	
PM2	Particulate Matter 2	instant	ug/m^3	REAL	
PM3	Particulate Matter 3	instant	ug/m^3	REAL	
PM4	Particulate Matter 4	instant	ug/m^3	REAL	
Pr	Pressure	instant	bar	REAL	
Q	Reactive Power	instant	VAr	REAL	
Q1	Reactive Power 1	instant	VAr	REAL	
Q2	Reactive Power 2	instant	VAr	REAL	
Q3	Reactive Power 3	instant	VAr	REAL	
Qmax	Reactive Power Max	instant	VAr	REAL	
Qmax1	Reactive Power 1 Max	instant	VAr	REAL	
Qmax2	Reactive Power 2 Max	instant	VAr	REAL	
Qmax3	Reactive Power 3 Max	instant	VAr	REAL	
R	Rate	instant	/h	REAL	
RAD	Radon	instant	Bq/m3	REAL	
RES	Resistance	instant	Ohms	REAL	
REr	Reactive Energy Relative	instant	VArh	REAL	
REr1	Reactive Energy 1 Relative	instant	VArh	REAL	

REr2	Reactive Energy 2 Relative	instant	VArh	REAL	
REr3	Reactive Energy 3 Relative	instant	VArh	REAL	
RExr	Reactive Energy Export Relative	instant	VArh	REAL	
RExr1	Reactive Energy Export 1 Relative	instant	VArh	REAL	
RExr2	Reactive Energy Export 2 Relative	instant	VArh	REAL	
RExr3	Reactive Energy Export 3 Relative	instant	VArh	REAL	
RH	Relative Humidity	instant	%	DOUBLE PRECISION	
S	Apparent Power	instant	VA	REAL	
S1	Apparent Power 1	instant	VA	REAL	
S2	Apparent Power 2	instant	VA	REAL	
S3	Apparent Power 3	instant	VA	REAL	
Smax	Apparent Power Max	instant	VA	REAL	
Smax1	Apparent Power 1 Max	instant	VA	REAL	
Smax2	Apparent Power 2 Max	instant	VA	REAL	
Smax3	Apparent Power 3 Max	instant	VA	REAL	
SO	Sound	instant	dB	REAL	
T	Temperature	instant	^{o}C	REAL	
T1	Temperature 1	instant	^{o}C	REAL	
T2	Temperature 2	instant	^{o}C	REAL	
T3	Temperature 3	instant	^{o}C	REAL	
T4	Temperature 4	instant	^{o}C	REAL	
T5	Temperature 5	instant	^{o}C	REAL	
T6	Temperature 6	instant	^{o}C	REAL	
T7	Temperature 7	instant	^{o}C	REAL	
T8	Temperature 8	instant	^{o}C	REAL	
Ti	Internal Temperature	instant	^{o}C	REAL	
Te	External Temperature	instant	^{o}C	REAL	
Tf	Flow Temperature	instant	^{o}C	BIGINT	
Tr	Return Temperature	instant	^{o}C	BIGINT	
Td	Temperature Difference	instant	^{o}C	BIGINT	
U	Voltage (LL)	instant	V	REAL	

U1	Voltage 1 (LL)	instant	V	REAL	
U2	Voltage 2 (LL)	instant	V	REAL	
U3	Voltage 3 (LL)	instant	V	REAL	
Umax	Voltage (LL) Max	instant	V	REAL	
Umax1	Voltage 1 (LL) Max	instant	V	REAL	
Umax2	Voltage 2 (LL) Max	instant	V	REAL	
Umax3	Voltage 3 (LL) Max	instant	V	REAL	
V	Voltage (LN)	instant	V	DOUBLE PRECISION	
V1	Voltage 1 (LN)	instant	V	REAL	
V2	Voltage 2 (LN)	instant	V	REAL	
V3	Voltage 3 (LN)	instant	V	REAL	
Vmax	Maximum System Voltage	instant	V	REAL	
Vmax1	Voltage 1 (LN) Max	instant	V	REAL	
Vmax2	Voltage 2 (LN) Max	instant	V	REAL	
Vmax3	Voltage 3 (LN) Max	instant	V	REAL	
btryVoltage	Battery Voltage	instant	V	DOUBLE PRECISION	
Vf	Volume Flow	instant	m^3/h	REAL	
Vfr	Volume Flow Relative	instant	m^3/h	REAL	
VOC1	Volatile Organic Compounds 1	instant	ppb	REAL	
VOC2	Volatile Organic Compounds 2	instant	ppb	REAL	
Med	Medium	instant		DOUBLE PRECISION	

Example Request

```
1   POST /dataingress/generic
2   [
3     {
4       "ts": 1707836800,
5       "uuId": "060000000F000700",
6       "period": 60,
7       "samples": 1,
8       "A12": 43.72565343218188,
9       "A13": 0.7187920625874736,
10      "AC": 15.924164799938056,
11      "AE": 1.6605267774595538,
12      "AE1": 58.0281068375465,
13      "AE2": 21.036792882269616,
14      "AE3": 55.48631677193862,
15      "AEr": 98.38190078660003,
16      "AEr1": 3.843108706103223,
```

```
17        "AEr2": 54.0953092528951,
18        "AEr3": 94.026905807679,
19        "AF": 25.0088909757365,
20        "AIN1": 17.712364447169175,
21        "AIN2": 19.59761964138863,
22        "AIN3": 56.63593805238415,
23        "AIN4": 15.230093717216558,
24        "AIN5": 54.10114249871165,
25        "AIN6": 44.6412468536949,
26        "AIN7": 25.707769778421042,
27        "AIN8": 17.46165832386429,
28        "AO1": 2.6744416461672826,
29        "AO2": 49.49436571891157,
30        "AO3": 50.28736537801445,
31        "AO4": 96.99234007717685,
32        "AO5": 18.530229673967806,
33        "AO6": 74.55853385597399,
34        "AO7": 16.097435455814672,
35        "AO8": 68.60785082383447,
36        "AP": 5.746747230061678,
37        "BTU": 75.86548695148224,
38        "BTUr": 55.95158578639459,
39        "btryVoltage": 92.61950445017754,
40        "C": 74.32616598639925,
41        "CO": 51.884906297496016,
42        "CO2": 26.49183861282274,
43        "Cr": 80.29194193905775,
44        "Ct": 42.49282001633794,
45        "D": 96.38148899952951,
46        "D1": 93.2981527844901,
47        "D2": 18.38801820687392,
48        "D3": 25.26024548581813,
49        "Dmax": 5.371356292335017,
50        "Dmax1": 30.704527502276246,
51        "Dmax2": 63.48037573670986,
52        "Dmax3": 84.59015064155224,
53        "DI1": 56.411229100269836,
54        "DI2": 4.756578665579007,
55        "DI3": 32.83214714975662,
56        "DI4": 5.382208808037549,
57        "DI5": 48.780601540943884,
58        "DI6": 48.660679510170915,
59        "DI7": 68.74997688864822,
60        "DI8": 88.89305227850241,
61        "DIt1": 65.89860071702796,
62        "DIt2": 0.15932098038462966,
63        "DIt3": 61.39424795349792,
64        "DIt4": 54.30660660113516,
65        "DIt5": 19.619053238825433,
66        "DIt6": 9.835229632367952,
67        "DIt7": 57.653330619285505,
68        "DIt8": 23.85571888827519,
69        "DIC1": 70.58439029873553,
70        "DIC2": 10.499148029135142,
71        "DIC3": 27.97981539181425,
72        "DIC4": 2.704013093702501,
73        "DIC5": 37.72645492000806,
74        "DIC6": 59.88552684890364,
```

```
75        "DIC7": 58.76979625726575,
76        "DIC8": 31.208877512816745,
77        "DICr1": 16.68503881277751,
78        "DICr2": 48.675271099747754,
79        "DICr3": 26.51802009275091,
80        "DICr4": 74.08466370500841,
81        "DICr5": 68.77678947931938,
82        "DICr6": 96.64123570390103,
83        "DICr7": 77.05784997765805,
84        "DICr8": 83.98593253641666,
85        "DO1": 43.42521347265002,
86        "DO2": 40.18181331464174,
87        "DO3": 65.30846132212712,
88        "DO4": 82.75740914175167,
89        "DO5": 20.38143548200905,
90        "DO6": 62.31412019688363,
91        "DO7": 5.177164313000238,
92        "DO8": 74.54497291452483,
93        "DOt1": 96.93287443226802,
94        "DOt2": 66.52555180086081,
95        "DOt3": 28.293297592687093,
96        "DOt4": 75.66094858369834,
97        "DOt5": 65.58058264925172,
98        "DOt6": 61.950350302248424,
99        "DOt7": 16.018129194163777,
100       "DOt8": 99.28224920261755,
101       "E": 51.04449645199091,
102       "E1": 71.49043477675431,
103       "E2": 54.21464576116514,
104       "E3": 85.73404279804511,
105       "Ex": 88.1606521961096,
106       "Ex1": 86.8344282670107,
107       "Ex2": 10.57068254359564,
108       "Ex3": 1.7137464609526782,
109       "Er": 20.206666886902724,
110       "Er1": 12.515026243643382,
111       "Er2": 47.295132115155056,
112       "Er3": 0.8291073613004327,
113       "Exr": 36.32237004876247,
114       "Exr1": 28.30536934933875,
115       "Exr2": 60.841362905149055,
116       "Exr3": 82.66140682746722,
117       "F": 3.406081489942075,
118       "I": 22.57731748864861,
119       "I1": 82.44367413336582,
120       "I2": 49.749529338325154,
121       "I3": 15.587723169283812,
122       "Imax": 2.017212008134095,
123       "Imax1": 54.89336636610952,
124       "Imax2": 44.94105989343536,
125       "Imax3": 64.63353413372931,
126       "In": 75.73413284296828,
127       "Inmax": 87.13520974253082,
128       "LL": 37.96845969649426,
129       "M": 4.334635149843355,
130       "Med": 65.33817465665666,
131       "Mf": 56.96488612190116,
132       "Mfr": 98.36175045853562,
```

```
133        "NOX1": 34.40497132688992,
134        "NOX2": 21.37382050108808,
135        "P": 75.09535596477583,
136        "P1": 53.65321769083534,
137        "P2": 20.134688783499733,
138        "P3": 4.094986992001061,
139        "Pmax": 44.70035077291557,
140        "Pmax1": 88.68327633882093,
141        "Pmax2": 17.281604659409076,
142        "Pmax3": 53.86099245113367,
143        "PE": 89.08517299643027,
144        "PEC": 95.82453593417281,
145        "PECR": 92.60919643221851,
146        "PF": 69.7500067622168,
147        "PF1": 27.058717993581073,
148        "PF2": 86.46952248479684,
149        "PF3": 23.549089312343437,
150        "PFmax": 1.6214862007747806,
151        "PFmax1": 21.60963389166148,
152        "PFmax2": 17.28383666709244,
153        "PFmax3": 32.03058304825359,
154        "PIR": 27.55479567104708,
155        "PIRr": 76.24630605627145,
156        "PIRt": 84.53742055433682,
157        "PM1": 36.17956723839956,
158        "PM2": 25.63641356566754,
159        "PM3": 35.26264658908483,
160        "PM4": 91.85935901098856,
161        "Pr": 15.766528302694919,
162        "Q": 73.87067581241517,
163        "Q1": 72.0623772926919,
164        "Q2": 83.8519372902121,
165        "Q3": 71.49415433942067,
166        "Qmax": 44.41753138993752,
167        "Qmax1": 13.932283461993691,
168        "Qmax2": 63.47002967422364,
169        "Qmax3": 78.26883079403491,
170        "R": 78.40242058895642,
171        "RAD": 81.79634273135864,
172        "RES": 11.540811886797107,
173        "RE": 68.17501479209169,
174        "RE1": 77.73186349204363,
175        "RE2": 36.56600994829368,
176        "RE3": 47.43953554306158,
177        "REx": 25.20488562304754,
178        "REx1": 3.0565457898455417,
179        "REx2": 47.452888054518446,
180        "REx3": 29.57410944140242,
181        "REr": 52.28609943403215,
182        "REr1": 39.724183985835026,
183        "REr2": 7.888353526540265,
184        "REr3": 72.59975516824039,
185        "RExr": 60.419989777924485,
186        "RExr1": 25.561986875516357,
187        "RExr2": 96.2193119787701,
188        "RExr3": 28.252020957065753,
189        "RH": 31.923050122299724,
190        "S": 3.554866464601302,
```

```
191        "S1": 84.63148678868613,
192        "S2": 70.14606570366121,
193        "S3": 8.993367296174805,
194        "Smax": 43.598927205241715,
195        "Smax1": 18.160494378842643,
196        "Smax2": 80.49097726144407,
197        "Smax3": 87.82655903502673,
198        "SO": 97.97640796656553,
199        "Stat": 82.17105780829259,
200        "T": 62.268263409970444,
201        "T1": 84.33233061075785,
202        "T2": 37.260887602931305,
203        "T3": 91.90671885940559,
204        "T4": 0.5240156783368511,
205        "T5": 39.89568964573354,
206        "T6": 35.47122098294609,
207        "T7": 65.88324385969119,
208        "T8": 69.393944958874,
209        "Ton": 67.97127880527232,
210        "Top": 18.49845369276519,
211        "Toff": 2.5116012909038,
212        "Trun": 4.721179327332033,
213        "Ti": 28.717259377575132,
214        "Te": 99.76326711464826,
215        "Tf": 14.358817839230792,
216        "Tr": 50.29431928428557,
217        "Td": 21.220333325313558,
218        "U": 40.777098872129386,
219        "U1": 89.42732940866954,
220        "U2": 62.55862850814994,
221        "U3": 9.086507609619995,
222        "Umax": 3.5561495477129474,
223        "Umax1": 94.2750221091672,
224        "Umax2": 23.52607144207045,
225        "Umax3": 27.861288854787723,
226        "V": 34.251338818227566,
227        "V1": 92.9640357815493,
228        "V2": 24.389187164785895,
229        "V3": 67.85032528818135,
230        "Vmax": 26.58195957847962,
231        "Vmax1": 30.099875261122307,
232        "Vmax2": 52.0401052907296,
233        "Vmax3": 59.197541912643956,
234        "Vf": 3.2089570552152384,
235        "Vfr": 85.24703508487299,
236        "VOC1": 55.300390606420294,
237        "VOC2": 30.41729840003759,
238        "Vol": 13.670207426729707
239      },
240      {
241        "ts": 1707836860,
242        "uuId": "060000000F000700",
243        "period": 60,
244        "samples": 1,
245        "A12": 44.72565343218188,
246        "A13": 0.8187920625874736,
247        "AC": 16.924164799938056,
248        "AE": 2.6605267774595538,
```

```json
      "AE1": 59.0281068375465,
      "AE2": 22.036792882269616,
      "AE3": 56.48631677193862,
      "AEr": 99.38190078660003,
      "AEr1": 4.843108706103223,
      "AEr2": 55.0953092528951,
      "AEr3": 95.026905807679,
      "AF": 26.0088909757365,
      "AIN1": 18.712364447169175,
      "AIN2": 20.597619641384863,
      "AIN3": 57.63593805238415,
      "AIN4": 16.230093717216558,
      "AIN5": 55.10114249871165,
      "AIN6": 45.6412468536949,
      "AIN7": 26.707769778421042,
      "AIN8": 18.46165832386429,
      "AO1": 3.6744416461672826,
      "AO2": 50.49436571891157,
      "AO3": 51.28736537801445,
      "AO4": 97.99234007717685,
      "AO5": 19.530229673967806,
      "AO6": 75.55853385597399,
      "AO7": 17.097435455814672,
      "AO8": 69.60785082383447,
      "AP": 6.746747230061678,
      "BTU": 76.86548695148224,
      "BTUr": 56.95158578639459,
      "btryVoltage": 93.61950445017754,
      "C": 75.32616598639925,
      "CO": 52.884906297496016,
      "CO2": 27.49183861282274,
      "Cr": 81.29194193905775,
      "Ct": 43.49282001633794,
      "D": 97.38148899952951,
      "D1": 94.2981527844901,
      "D2": 19.38801820687392,
      "D3": 26.26024548581813,
      "Dmax": 6.371356292335017,
      "Dmax1": 31.704527502276246,
      "Dmax2": 64.48037573670986,
      "Dmax3": 85.59015064155224,
      "DI1": 57.411229100269836,
      "DI2": 5.756578665579007,
      "DI3": 33.83214714975662,
      "DI4": 6.382208808037549,
      "DI5": 49.780601540943884,
      "DI6": 49.660679510170915,
      "DI7": 69.74997688864822,
      "DI8": 89.89305227850241,
      "DIt1": 66.89860071702796,
      "DIt2": 0.25932098038462966,
      "DIt3": 62.39424795349792,
      "DIt4": 55.30660660113516,
      "DIt5": 20.619053238825433,
      "DIt6": 10.835229632367952,
      "DIt7": 58.65333060613285505,
      "DIt8": 24.85571888827519,
      "DIC1": 71.58439029873553,
```

```
307        "DIC2": 11.499148029135142,
308        "DIC3": 28.97981539181425,
309        "DIC4": 3.704013093702501,
310        "DIC5": 38.72645492000806,
311        "DIC6": 60.88552684890364,
312        "DIC7": 59.76979625726575,
313        "DIC8": 32.208877512816745,
314        "DICr1": 17.68503881277751,
315        "DICr2": 49.675271099747754,
316        "DICr3": 27.51802009275091,
317        "DICr4": 75.08466370500841,
318        "DICr5": 69.77678947931938,
319        "DICr6": 97.64123570390103,
320        "DICr7": 78.05784997765805,
321        "DICr8": 84.98593253641666,
322        "DO1": 44.42521347265002,
323        "DO2": 41.18181331464174,
324        "DO3": 66.30846132212712,
325        "DO4": 83.75740914175167,
326        "DO5": 21.38143548200905,
327        "DO6": 63.31412019688363,
328        "DO7": 6.177164313000238,
329        "DO8": 75.54497291452483,
330        "DOt1": 97.93287443226802,
331        "DOt2": 67.52555180086081,
332        "DOt3": 29.293297592687093,
333        "DOt4": 76.66094858369834,
334        "DOt5": 66.58058264925172,
335        "DOt6": 62.950350302248424,
336        "DOt7": 17.018129194163777,
337        "DOt8": 100.28224920261755,
338        "E": 52.04449645199091,
339        "E1": 72.49043477675431,
340        "E2": 55.21464576116514,
341        "E3": 86.73404279804511,
342        "Ex": 87.1606521961096,
343        "Ex1": 87.8344282670107,
344        "Ex2": 11.57068254359564,
345        "Ex3": 2.7137464609526782,
346        "Er": 21.206666886902724,
347        "Er1": 13.515026243643382,
348        "Er2": 48.295132115155056,
349        "Er3": 1.8291073613004327,
350        "Exr": 37.32237004876247,
351        "Exr1": 29.30536934933875,
352        "Exr2": 61.841362905149055,
353        "Exr3": 83.66140682746722,
354        "F": 4.406081489942075,
355        "I": 23.57731748864861,
356        "I1": 83.44367413336582,
357        "I2": 51.749529338325154,
358        "I3": 16.587723169283812,
359        "Imax": 3.017212008134095,
360        "Imax1": 55.89336636610952,
361        "Imax2": 45.94105989343536,
362        "Imax3": 65.63353413372931,
363        "In": 76.73413284296828,
364        "Inmax": 88.13520974253082,
```

```
365        "LL": 38.96845969649426,
366        "M": 5.334635149843355,
367        "Med": 66.33817465665666,
368        "Mf": 57.96488612190116,
369        "Mfr": 99.36175045853562,
370        "NOX1": 35.40497132688992,
371        "NOX2": 22.37382050108808,
372        "P": 76.09535596477583,
373        "P1": 54.65321769083534,
374        "P2": 21.134688783499733,
375        "P3": 5.094986992001061,
376        "Pmax": 45.70035077291557,
377        "Pmax1": 89.68327633882093,
378        "Pmax2": 18.281604659409076,
379        "Pmax3": 54.86099245113367,
380        "PE": 90.08517299643027,
381        "PEC": 96.82453593417281,
382        "PECR": 93.60919643221851,
383        "PF": 70.7500067622168,
384        "PF1": 28.058717993581073,
385        "PF2": 87.46952248479684,
386        "PF3": 24.549089312343437,
387        "PFmax": 2.6214862007747806,
388        "PFmax1": 22.60963389166148,
389        "PFmax2": 18.28383666709244,
390        "PFmax3": 33.03058304825359,
391        "PIR": 28.55479567104708,
392        "PIRr": 77.24630605627145,
393        "PIRt": 85.53742055433682,
394        "PM1": 37.17956723839956,
395        "PM2": 26.63641356566754,
396        "PM3": 36.26264658908483,
397        "PM4": 92.85935901098856,
398        "Pr": 16.766528302694919,
399        "Q": 74.87067581241517,
400        "Q1": 73.0623772926919,
401        "Q2": 84.8519372902121,
402        "Q3": 72.49415433942067,
403        "Qmax": 45.41753138993752,
404        "Qmax1": 14.932283461993691,
405        "Qmax2": 64.47002967422364,
406        "Qmax3": 79.26883079403491,
407        "R": 79.40242058895642,
408        "RAD": 82.79634273135864,
409        "RES": 12.540811886797107,
410        "RE": 69.17501479209169,
411        "RE1": 78.73186349204363,
412        "RE2": 37.56600994829368,
413        "RE3": 48.43953554306158,
414        "REx": 26.20488562304754,
415        "REx1": 4.0565457898455417,
416        "REx2": 48.452888054518446,
417        "REx3": 30.57410944140242,
418        "REr": 53.28609943403215,
419        "REr1": 40.724183985835026,
420        "REr2": 8.888353526540265,
421        "REr3": 73.59975516824039,
422        "RExr": 61.419989777924485,
```

```
423        "RExr1": 26.561986875516357,
424        "RExr2": 97.2193119787701,
425        "RExr3": 29.252020957065753,
426        "RH": 32.923050122299724,
427        "S": 4.554866464601302,
428        "S1": 85.63148678868613,
429        "S2": 71.14606570366121,
430        "S3": 9.993367296174805,
431        "Smax": 44.598927205241715,
432        "Smax1": 19.160494378842643,
433        "Smax2": 81.49097726144407,
434        "Smax3": 88.82655903502673,
435        "SO": 98.97640796656553,
436        "Stat": 83.17105780829259,
437        "T": 63.268263409970444,
438        "T1": 85.33233061075785,
439        "T2": 38.260887602931305,
440        "T3": 92.90671885940559,
441        "T4": 1.5240156783368511,
442        "T5": 40.89568964573354,
443        "T6": 36.47122098294609,
444        "T7": 66.88324385969119,
445        "T8": 70.393944958874,
446        "Ton": 68.97127880527232,
447        "Top": 19.49845369276519,
448        "Toff": 3.5116012909038,
449        "Trun": 5.721179327332033,
450        "Ti": 29.717259377575132,
451        "Te": 100.76326711464826,
452        "Tf": 15.358817839230792,
453        "Tr": 51.29431928428557,
454        "Td": 22.220333325313558,
455        "U": 41.777098872129386,
456        "U1": 90.42732940866954,
457        "U2": 63.55862850814994,
458        "U3": 10.086507609619995,
459        "Umax": 4.5561495477129474,
460        "Umax1": 95.2750221091672,
461        "Umax2": 24.52607144207045,
462        "Umax3": 28.861288854787723,
463        "V": 35.251338818227566,
464        "V1": 93.9640357815493,
465        "V2": 25.389187164785895,
466        "V3": 68.85032528818135,
467        "Vmax": 27.58195957847962,
468        "Vmax1": 31.099875261122307,
469        "Vmax2": 53.0401052907296,
470        "Vmax3": 60.197541912643956,
471        "Vf": 4.2089570552152384,
472        "Vfr": 86.24703508487299,
473        "VOC1": 56.300390606420294,
474        "VOC2": 31.417298400037595,
475        "Vol": 14.670207426729707
476     }
477  ]
```

Custom Parameters

Custom parameters are an extension to the generic data ingress API that allows Users to upload production type data relevant for their use case, for example number of "widgets" produced per hour, amount of waste material produced per day, etc.

These can optionally be used in conjuntion with the general parameters

> ℹ It is possible to have up to 12 custom paramters associated with a generic device.

> ℹ In order to upload data to a custom parameter it must have been previously created and associated with a generic meter.

The following is an example request to upload to a custom paramter called "Pmph"

```
POST /dataingress/generic
[
  {
    "ts": 1707836800,
    "uuId": "060000000F000700",
    "period": 60,
    "samples": 1,
    "customParameters":
      {
        "Pmph": 27
      }
  },
  {
    "ts": 1707836860,
    "uuId": "060000000F000700",
    "period": 60,
    "samples": 1,
     "customParameters":
       {
         "Pmph": 23
       }
  }
]
```

GET /devicecustomparameters

Retrieve a list of custom parameters associated with a specific device

`GET` `GET /devicecustomparameters`

Parameters

deviceId	Required	Filter by device ID.

GET Response Body

The response is in JSON format and includes the following fields:

Field	Data Type	Description
id	int	The unique identifier for the device-to-custom-parameter relationship.
deviceId	int	The identifier of the device.
customParameterID	string	The identifier of the custom parameter.
createdBy	int	The identifier of the user who created the account.
createdTs	string	The timestamp of when the account was created.

```
1   {
2       "result": [
3           {
4               "id": "1",
5               "deviceId": "23161",
6               "customParameterId": "1",
7               "createdBy": "13904",
8               "createdTs": "1713265169"
9           },
10          {
11              "id": "2",
12              "deviceId": "23161",
13              "customParameterId": "2",
14              "createdBy": "13904",
15              "createdTs": "1713324559"
16          },
17          {
18              "id": "3",
19              "deviceId": "23161",
20              "customParameterId": "3",
21              "createdBy": "13904",
22              "createdTs": "1713324713"
```

```
23          },
24          {
25              "id": "4",
26              "deviceId": "23161",
27              "customParameterId": "3",
28              "createdBy": "13904",
29              "createdTs": "1713324755"
30          }
31      ],
32      "meta": {
33          "currentPage": 1,
34          "recordsPerPage": 20,
35          "recordCount": 4,
36          "pageCount": 1
37      }
38  }
```

Retrieve a specific device-to-custom-parameter relationship by its ID

`GET` GET /meters/{deviceCustomParameterId}

Parameters

`deviceCustomPa rameterId`	Required	The unique identifier of the device-to-custom-parameter relationship.

GET /devices

Devices refer to a physical piece of equipment such as an Eniscope or a metering point on a physical device. Devices can have a number of metering points attached to them these can be found under the meteringPoints property.

Retrieve a list of devices

GET `GET /devices`

Parameters

These parameters can be used to filter and refine the results when making API calls to retrieve device information.

`name`	Optional	Filter by device name.
`parent`	Optional	Filter by parent device ID.
`device`	Optional	Filter by device ID.
`organization`	Optional	Filter by organization ID.
`uuid`	Optional	Filter by device UUID (including extra digits).
`registered`	Optional	Filter by registration status.
`expires`	Optional	Filter by expiration date.
`renewed`	Optional	Filter by renewal date.
`deviceType`	Optional	Filter by device type ID.
`ftpuser`	Optional	Filter by FTP user ID.
`deviceTypeId`	Optional	Filter by device type ID.
`vendor`	Optional	Filter by vendor name.
`vendor_id`	Optional	Filter by vendor ID.

Example Request

```
1  GET /devices?organization=123&name=device1&uuid=0600007777B80000&vendor_id=456
```

Retrieve a specific device by ID

GET `GET /devices/{deviceId}`

Example Request

```
1  GET /devices/{deviceId}
```

GET Response Body

The response is in JSON format and includes the following fields:

Field	Data Type	Description
`deviceId`	int	Unique identifier for the device.
`deviceTypeId`	int	Identifier for the device type.
`deviceTypeCode`	string	Code for the device type.
`deviceTypeName`	string	Name of the device type.
`nominalCode`	string or null	Nominal code associated with the device.
`meteringPoint`	int	Identifier for the metering point.
`singlePhaseEnabled`	int	Indicates if single-phase is enabled (1 for true).
`createdBy`	int	Identifier of the user who created the device.
`modifiedBy`	int or null	Identifier of the user who last modified the device.
`createdTs`	string	Timestamp of when the device was created.
`modifiedTs`	string or null	Timestamp of when the device was last modified.
`format_definition`	string	Device format definition.
`timeZone`	string	Timezone of the device.
`uuId`	string	UUID of the device
`parentId`	int	Identifier of the parent device (or null).
`organizationId`	int	Identifier of the organization associated with it.
`subscriptionId`	null	Subscription identifier (null).
`binaryStoreId`	int	Identifier for the binary store.
`serviceAgreementId`	null	Service agreement identifier (null).
`deviceName`	string	Name of the device
`registered`	string	Timestamp when the device was registered.
`dataPlan`	string	Data plan identifier.
`expires`	string	Expiration date timestamp.
`renewed`	string	Timestamp when the device was renewed.
`status`	int	Device status (1 for active).
`vendor`	string	Vendor name.
`vendor_id`	null	Vendor identifier (null).

`vendor_format_id`	null	Vendor format identifier (null).
`vendor_parent_id`	string	Vendor parent identifier
`txm`	null	TXM identifier (null).
`sortCode`	string	Sort code associated with the device
`isMeteringPoint`	boolean	Indicates if it's a metering point (true).
`meteringPoints`	array	Array of metering point details.
`tempteratureEnabled`	int	Indicates if temperature is enabled (1 for true).

```json
{
    "deviceId": 500,
    "links": {
        "self": "https://core.eniscope.com/v1/1/devices/500",
        "repository": "https://core.eniscope.com/v1/1/devices/500/repository",
        "services": "https://core.eniscope.com/v1/1/devices/500/services",
        "status": "https://core.eniscope.com/v1/1/devices/500/status",
        "readings": "https://core.eniscope.com/v1/1/devices/500/readings",
        "verification": "https://core.eniscope.com/v1/1/devices/500/verification",
        "events": "https://core.eniscope.com/v1/1/devices/500/events",
        "treadlines": "https://core.eniscope.com/v1/1/devices/500/treadlines",
        "interfaces": "https://core.eniscope.com/v1/1/devices/500/interfaces"
    },
    "deviceTypeId": 4,
    "deviceTypeCode": "EM_1",
    "deviceTypeName": "Eniscope 8 Metering Point",
    "nominalCode": null,
    "meteringPoint": 1,
    "singlePhaseEnabled": 1,
    "createdBy": 859,
    "modifiedBy": 14824,
    "createdTs": "1397546058",
    "modifiedTs": "1648818678",
    "format_definition": "",
    "timeZone": "Europe/London",
    "uuId": "0000000000000000",
    "parentId": 2587,
    "organizationId": 11264,
    "subscriptionId": null,
    "binaryStoreId": 2,
    "serviceAgreementId": null,
    "deviceName": "Example Device Name",
    "registered": "1397546058",
    "dataPlan": "12",
    "expires": "1679702400",
    "renewed": "0",
    "status": 1,
    "vendor": "",
    "vendor_id": null,
    "vendor_format_id": null,
    "vendor_parent_id": "",
    "txm": null,
    "sortCode": "1-2-3-18518-11313-11264-",
    "isMeteringPoint": true,
```

```
45      "meteringPoints": [
46          {
47              "deviceId": 500,
48              "uuId": "0000000000000000",
49              "parentId": 2587,
50              "organizationId": 11264,
51              "subscriptionId": null,
52              "deviceTypeId": 4,
53              "binaryStoreId": 2,
54              "serviceAgreementId": null,
55              "deviceName": "Example Device Name",
56              "registered": "1397546058",
57              "dataPlan": "12",
58              "expires": "1679702400",
59              "renewed": "0",
60              "nominalCode": null,
61              "status": 1,
62              "createdBy": 859,
63              "modifiedBy": 6509,
64              "createdTs": "1397477385",
65              "modifiedTs": "1502198399",
66              "vendor": "",
67              "vendor_id": null,
68              "vendor_format_id": null,
69              "vendor_parent_id": "",
70              "txm": null,
71              "meterId": 500,
72              "deviceTypeInterfaceId": 12,
73              "dataType": "grid meter",
74              "ct": "",
75              "vs": "",
76              "channels": 3,
77              "pulseLabel": "",
78              "pulseRatio": "1",
79              "pulseUnits": "",
80              "pulseEnergyRatio": "0",
81              "temperatureEnabled": 1,
82              "dataReceivedTs": "1487053045",
83              "isLeadingTrailing": "0",
84              "sortCode": "1-2-3-18518-11313-11264-",
85              "interface": {
86                  "deviceTypeInterfaceId": 12,
87                  "links": {
88                      "self": "https://core.eniscope.com/v1/1/deviceTypeInterfaces/12"
89                  },
90                  "interfaceName": "Onboard",
91                  "deviceTypeId": 7,
92                  "deviceInterfaceId": 3,
93                  "direction": "accept",
94                  "count": 8,
95                  "prefix": "00",
96                  "createdBy": 1,
97                  "modifiedBy": 1,
98                  "createdTs": "1417606561",
99                  "modifiedTs": "1417606561"
100                 }
101             }
102         ],
```

```
103        "tempteratureEnabled": 1
104    }
```

GET /devicetypes

Retrieve a list of device types

`GET` `GET /devicetypes`

Parameters

`deviceTypeCode`	Optional	Filter by device type code.
`deviceTypeName`	Optional	Filter by device type name.
`meteringPoint`	Optional	Filter by metering point status.
`singlePhaseEnabled`	Optional	Filter by single-phase enabled status.

Example Request:

```
1  GET /devicetypes?deviceTypeCode=123&deviceTypeName=TypeA&meteringPoint=true&singlePhaseEnabled=true
```

Retrieve a specific device type by its ID

`GET` `GET /devicetypes/{deviceTypeId}`

Parameters

`deviceTypeId`	Required	The unique identifier of the device type.

Example Request

```
1  GET /devicetypes/123
```

GET /events

Retrieve a list of events

`GET` `GET /events`

Parameters

`organization`	Optional	Filter by organization ID.
`resource`	Optional	Filter by resource ID.
`account`	Optional	Filter by account ID.
`name`	Optional	Filter by event name.
`description`	Optional	Filter by event description.
`type`	Optional	Filter by event type.
`startTs`	Optional	Filter by event start timestamp. Example where startTs is equal or greater than a time. startTs=>1630944000

Example Request

```
1  GET /events?
   organization=123&resource=456&account=789&name=EventA&description=DescriptionB&type=TypeX&startTs=1630944000
```

Retrieve a specific event by its ID

`GET` `GET /events/{eventId}`

Parameters

`eventId`	Required	The unique identifier of the event.

Example Request

```
1  GET /events/123
```

GET Response Body

Field	Data Type	Description
`eventId` (int)	Integer	The unique identifier for the event.
`eventName` (string)	String	The name or title of the event.

`eventDescription`	String	A detailed description of the event.
`resourceId` (int)	Integer	The identifier of the associated resource.
`accountId` (int)	Integer	The identifier of the account to which the event is related.
`eventType` (string)	String	The type or category of the event.
`eventStartTs` (string)	String	The timestamp when the event started.
`eventEndTs` (string)	String	The timestamp when the event ended.
`createdTs` (string)	String	The timestamp when the event was created.
`modifiedTs` (string)	String	The timestamp when the event was last modified.
`links` (object)	Object	Links to related resources.

```
1   {
2       "eventId": "2303",
3       "eventName": "Maintenance Check",
4       "eventDescription": "Routine maintenance check for HVAC system",
5       "resourceId": "127",
6       "accountId": "312",
7       "eventType": "Maintenance",
8       "eventStartTs": "1673568000",
9       "eventEndTs": "1673568100",
10      "createdTs": "1673496000",
11      "modifiedTs": "1673552000",
12      "links": {
13          "self": "https://core.eniscope.com/v1/events/2303",
14          "resource": "https://core.eniscope.com/v1/resources/127",
15          "account": "https://core.eniscope.com/v1/accounts/312"
16      }
17  }
```

GET /fields

Retrieve a list of all fields

GET `GET /fields`

Example Request

```
1   GET /fields
```

Retrieve fields available for a specific device type by its ID

GET `GET /fields/{deviceTypeId}`

Parameters

`deviceTypeId`	Required	The unique identifier of the device type.

Example Request

```
1   GET /fields/123
```

GET /meters

Retrieve a list of meters

`GET` `GET /meters`

Parameters

`name`	Optional	Filter by meter name.
`parent`	Optional	Filter by parent meter ID.
`device`	Optional	Filter by device ID.
`organizationNa me`	Optional	Filter by organization name.
`organization`	Optional	Filter by organization ID.
`uuid`	Optional	Filter by UUID.
`registered`	Optional	Filter by registration status.
`renewed`	Optional	Filter by renewal status.
`expires`	Optional	Filter by expiration status.
`dataType`	Optional	Filter by data type.

Example Request

```
1  GET /meters?
   name=MeterA&parent=123&device=456&organizationName=OrgX&organization=789&uuid=UUID123&registered=true&renewed=t
   rue&expires=false&dataType=TypeX
```

Retrieve a specific meter by its device ID

`GET` `GET /meters/{meterId}`

Parameters

`deviceId`	Required	The unique identifier of the meter.

Example Request

```
1  GET /meters/123
```

GET /organizations

Organizations in the Eniscope Platform are organized in a hierarchy. This hierarchy controls what is viewable by the logged-in user. A user can only view their own organization and the children of that organization this affects all aspects of the system.

Retrieve a list of organizations

To get a list of orgnanizations viewable by the logged-in user

organizations `GET` `GET /organizations`

Parameters

`id`	Optional	Filter by organization ID.
`name`	Optional	Filter by organization name.
`email`	Optional	Filter by default email address.
`parent`	Optional	Filter by parent organization ID.
`type`	Optional	Filter by organization type.

Example Request

```
1  GET /organizations?id=123&name=OrgX&email=email@example.com&parent=456&type=TypeX
```

Retrieve a specific organization by its ID

`GET` `GET /organizations/{organizationId}`

Parameters

`organizationId`	Required	The unique identifier of the organization.

Example Request

```
1  GET /organizations/123
```

GET /readings

The readings endpoint is the main way of querying meter data stored on the Eniscope Platform

Retrieve readings for a specific channel

`GET` `GET /readings/{channelId}`

Parameters

`channelId`	Required	The unique identifier of the channel.			
`fields[]`	Optional	A list of energy parameters such as E, V, I, etc., to be included in the response. Fields are always requested as an array e.g. `fields[]=E&fields[]=V` Some of the commonly requested fields are:			
		`E` - Energy (System)	`E1` - Energy (Phase 1)	`E2` - Energy (Phase 2)	`E3` - Energy (Phase 3)
		`I` - Current (System)	`I1` - Current (Phase 1)	`I2` - Current (Phase 2)	`I3` - Current (Phase 3)
		`P` - Power, (System)	`P1` - Power (Phase 1)	`P2` - Power (Phase 2)	`P3` - Power (Phase 3)
		`PF` - Power Factor (System)	`PF1` - Power Factor (Phase 1)	`PF2` - Power Factor (Phase 2)	`PF3` - Power Factor (Phase 3)
		`Q` - Reactive Power (System)	`Q1` - Reactive Power (Phase 1)	`Q2` - Reactive Power (Phase 2)	`Q3` - Reactive Power (Phase 3)
		`S` - Apparent Power (System)	`S1` - Apparent Power (Phase 1)	`S2` - Apparent Power (Phase 2)	`S3` - Apparent Power (Phase 3)
		`U` - Voltage LL (System)	`U1` - Voltage LL (Phase 1)	`U2` - Voltage LL (Phase 2)	`U3` - Voltage LL (Phase 3)
		`V` - Voltage LN (System)	`V1` - Voltage LN (Phase 1)	`V2` - Voltage LN (Phase 2)	`V3` - Voltage LN (Phase 3)
		`T` - Temperature (System)	`pulse` - Pulse count (System)	`cost` - if a tariff has been associated with a Channel/Dataview (System)	
		The list of available fields varies, depending on what the metering device can return. To get a list of fields associated with a device type use `GET /fields/{deviceType}` e.g. To get the list of available fields for an Eniscope Hybrid 8 use `GET /fields/20`			
`daterange`	Optional	A `daterange` can either be provided as an array of timestamps or as a pre-defined word describing the daterange			

Custom Date Ranges

Custom date ranges are requested by supplying the `daterange` as an array, containing the start timestamp and the end timestamp.

- The start time must be before the end time

The `daterange` timestamps can be defined in seconds in the Unix Epoch Timestamp format (e.g., 1694034000) which will provide granular date range selections

    e.g. `daterange[]=1694034000&daterange[]=1694062800`

Alternatively, the time stamps can be defined as "YYYY-MM-DD" which will provide granularity down to a day

    e.g. `daterange[]=2023-09-06&daterange[]=2023-09-07`

Pre-defined Ranges

The list of allowed pre-defined date ranges are:

- `today`
- `yesterday`
- `daybefore`
- `thisweek`
- `lastweek`
- `weekbefore`
- `7days` - last 7 days, from the current day
- `previous7days`
- `28days` - last 28 days, from the current day
- `30days` - last 30 days, from the current day
- `thismonth`
- `lastmonth`
- `thisyear`
- `lastyear`

e.g. `daterange=lastweek`

`res`	Optional	This is the resolution that the meter data will be summarized to. Valid resolution values are: - `auto` - `60` (1 minute) - `900` (15 minutes) - `1800` (30 minutes) - `3600` (1 hour) - `86400` (1 day) There are restrictions on that limit which resolutions can be selected based on the daterange selected.
`action`	Optional	The action parameter sets the kind of request that will be returned. Valid actions are: - `summarize` - returns a list of records at the resolution set in the request. - `total` - returns the total for each field request for the date range provided. - `averageday` - returns an average day profile for the date range provided. - `typicalday` - `medianday`

- `meanday`
- `minday`
- `maxday`

The default action type is `summarize`

`showCount ers`	Option al	A boolean to set how counter fields are returned. • If `showCounters` is set to true (1) then all counters will return the counter value • If `showCounters` is set to false (0) then it will calculate and return the usage or delta value for the `daterange`
`multiplie r`	Option al	This enables a multiplier to be applied to a field. For example to convert energy to BTU
units	Option al	Unit for the applied multiplier.
label	Option al	Label for the applied multiplier.
returnField	Option al	Boolean to return the original field value as well as the multiplied one.

Example Request

```
1  GET /readings/12345?fields[]=E&fields[]=V&daterange[]=1630965600&daterange[]=1633557600&res=3600&showCounters=1
```

GET /registrations

Retrieve device registrations based on specified parameters

`GET` `GET /registrations`

Parameters

deviceId	Optional	The unique identifier of the device.
uuid	Optional	The UUID of the device.
expired	Optional	Set to true or false to filter registrations by expiration status.

Example Request

```
1  GET /registrations?deviceId=12345&expired=true
```

Retrieve registrations for a specific device

`GET` `GET /registrations/{deviceId}`

Parameters

deviceId	Required	The unique identifier of the device.

Example Request

```
1  GET /registrations/12345
```

All Status Reponse Codes

Statuses from the API are provided as HTTP response statuses including any exceptions thrown by the API.

The following are all the status codes used by the Core API

Status Code	Description
100	Continue
101	Switching Protocols
102	Processing
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
207	Multi-status
208	Already Reported
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
306	Switch Proxy
307	Temporary Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed

406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-out
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
418	I\m a teapot
422	Unprocessable Entity
423	Locked
424	Failed Dependency
425	Unordered Collection
426	Upgrade Required
428	Precondition Required
429	Too Many Requests
431	Request Header Fields Too Large
451	Unavailable For Legal Reasons
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Time-out
505	HTTP Version not supported
506	Variant Also Negotiates
507	Insufficient Storage
508	Loop Detected
511	Network Authentication Required

Example PHP Code - API Request

The following is an example of a basic API request handler in PHP

```php
<?php
/*************************************************************************
Example Eniscope API Request Using PHP
(c) BEST Energy Saving Technology
*************************************************************************/

/**
 * Define which environment to send requests to
 *
 * https://core.eniscope.com - Live API URL
 */

const ENISCOPE_API_URL = 'https://core.eniscope.com';

/**
 * The endpoint to request
 *
 * For example:
 * /organizations
 * /readings/{channelId}/?action=summarise&daterange=today&fields[]=E&fields[]=E1&res=3600
 */

const ENDPOINT = '/';

/**
 * Eniscope Core authentication credentials
 *
 * ** Enter your details here
 */

const ENISCOPE_API_KEY = '';
const ENISCOPE_USERNAME = '';
const ENISCOPE_PASSWORD = '';

/**
 * Start PHP session handler
 * (We are using sessions to store the returned Eniscope Authentication Token)
 */

session_start();

/**
 * Create an array of CURL options for the request
 */

$curlOptions = array(
    CURLOPT_URL => ENISCOPE_API_URL . ENDPOINT, // Set the full URL to request
    CURLOPT_RETURNTRANSFER => 1, // Return the response
```

```php
 49      CURLOPT_HEADER => 1, // Also return the headers
 50      CURLOPT_HTTPHEADER => array(
 51          'X-Eniscope-API: ' . ENISCOPE_API_KEY
 52      ) // add the API Key header
 53  );
 54
 55  /**
 56   * Handle authentication for this request
 57   * The first request we make needs to use basic HTTPAuth with the user's
 58   * username and password.
 59   * If successful, then an Eniscope-Token is returned which is used on subsequent requests
 60   */
 61
 62  if (isset($_SESSION['Eniscope_Session_Token'])) {
 63      $curlOptions[CURLOPT_HTTPHEADER][] = "X-Eniscope-Token: {$_SESSION['Eniscope_Session_Token']}";
 64  } else {
 65      $curlOptions[CURLOPT_USERPWD] = ENISCOPE_USERNAME . ':' . md5(ENISCOPE_PASSWORD);
 66  }
 67
 68  /**
 69   * Perform the CURL request
 70   */
 71
 72  $curlRequest = curl_init();
 73  curl_setopt_array($curlRequest, $curlOptions);
 74  $fullResponse = curl_exec($curlRequest);
 75
 76  // Grab the HTTP response code
 77  $httpResponseCode = curl_getinfo($curlRequest, CURLINFO_HTTP_CODE);
 78
 79  // Extract the headers from the response
 80  $headerLength = curl_getinfo($curlRequest, CURLINFO_HEADER_SIZE);
 81  $headers      = substr($fullResponse, 0, $headerLength);
 82
 83  // Extract the body content from the response
 84  $body = substr($fullResponse, $headerLength);
 85
 86  // Look for the X-Eniscope-Token in the header
 87  if (0 !== preg_match("/X-Eniscope-Token: (.*)/", $headers, $m)) {
 88      // It's found, this means authentication has been performed
 89      $_SESSION['Eniscope_Session_Token'] = $m[1];
 90  }
 91
 92  /**
 93   * The Eniscope REST API uses HTTP return codes to determine
 94   * whether a request was successful
 95   */
 96  if ($httpResponseCode < 300) {
 97      // Request was successful
 98      $data = json_decode($body, true); // Decode the JSON response into an array
 99      echo '<pre>';
100      print_r($data);
101      echo '</pre>';
102      exit;
103  } else {
104      // An error state was returned
105
106  ?>
```

66

```php
107
108     <h1>Request Error</h1>
109     <h2>HTTP Response Code</h2>
110     <p><?php
111         echo $httpResponseCode;
112     ?></p>
113     <h2>Headers</h2>
114     <pre><?php
115         print_r($headers);
116     ?></pre>
117     <h2>Body</h2>
118     <pre><?php
119         echo htmlentities($body);
120     ?></pre>
121     <?php
122         exit;
123 }
```

API Fields

API Field	Label	Type	Unit	Eniscope Register	Data Type
AE	Apparent Energy	counter	VAh	AE	DOUBLE PRECISION
AE1	Apparent Energy 1	counter	VAh	AE1	DOUBLE PRECISION
AE2	Apparent Energy 2	counter	VAh	AE2	DOUBLE PRECISION
AE3	Apparent Energy 3	counter	VAh	AE3	DOUBLE PRECISION
BTU	BTU Absolute	counter		BTU	REAL
C	Count	counter		C	REAL
Ct	Time Since Last Pulse	counter	s	C_t	REAL
DI1	Digital Input 1	instant		DI1	DOUBLE PRECISION
DIt1	Time Since Last Input Event 1	instant	s	DI_t1	DOUBLE PRECISION
DIC1	Digital Input Counter 1	counter		DI_C1	DOUBLE PRECISION
DI2	Digital Input 2	instant		DI2	DOUBLE PRECISION
DIt2	Time Since Last Input Event 2	instant	s	DI_t2	DOUBLE PRECISION
DIC2	Digital Input Counter 2	counter		DI_C2	DOUBLE PRECISION
DI3	Digital Input 3	instant		DI3	DOUBLE PRECISION
DIt3	Time Since Last Input Event 3	instant	s	DI_t3	DOUBLE PRECISION
DIC3	Digital Input Counter 3	counter		DI_C3	DOUBLE PRECISION
DI4	Digital Input 4	instant		DI4	DOUBLE PRECISION
DIt4	Time Since Last Input Event 4	instant	s	DI_t4	DOUBLE PRECISION
DIC4	Digital Input Counter 4	counter		DI_C4	DOUBLE PRECISION
DI5	Digital Input 5	instant		DI5	REAL
DIt5	Time Since Last Input Event 5	instant	s	DI_t5	REAL
DIC5	Digital Input Counter 5	counter		DI_C5	REAL
DI6	Digital Input 6	instant		DI6	REAL
DIt6	Time Since Last Input Event 6	instant	s	DI_t6	REAL

DIC6	Digital Input Counter 6	counter		DI_C6	REAL
DI7	Digital Input 7	instant		DI7	REAL
DIt7	Time Since Last Input Event 7	instant	s	DI_t7	REAL
DIC7	Digital Input Counter 7	counter		DI_C7	REAL
DI8	Digital Input 8	instant		DI8	REAL
DIt8	Time Since Last Input Event 8	instant	s	DI_t8	REAL
DIC8	Digital Input Counter 8	counter		DI_C8	REAL
DO1	Digital Output 1	instant		DO1	DOUBLE PRECISION
DOt1	Time Since Last Output Event 1	instant		DO_t1	DOUBLE PRECISION
DO2	Digital Output 2	instant		DO2	DOUBLE PRECISION
DOt2	Time Since Last Output Event 2	instant		DO_t2	DOUBLE PRECISION
DO3	Digital Output 3	instant		DO3	DOUBLE PRECISION
DOt3	Time Since Last Output Event 3	instant		DO_t3	DOUBLE PRECISION
DO4	Digital Output 4	instant		DO4	REAL
DOt4	Time Since Last Output Event 4	instant		DO_t4	REAL
DO5	Digital Output 5	instant		DO5	REAL
DOt5	Time Since Last Output Event 5	instant		DO_t5	REAL
DO6	Digital Output 6	instant		DO6	REAL
DOt6	Time Since Last Output Event 6	instant		DO_t6	REAL
DO7	Digital Output 7	instant		DO7	REAL
DOt7	Time Since Last Output Event 7	instant		DO_t7	REAL
DO8	Digital Output 8	instant		DO8	REAL
DOt8	Time Since Last Output Event 8	instant		DO_t8	REAL
E	Energy	counter	Wh	E	DOUBLE PRECISION
E1	Energy 1	counter	Wh	E1	DOUBLE PRECISION
E2	Energy 2	counter	Wh	E2	REAL

E3	Energy 3	counter	Wh	E3	REAL
Ex	Export Energy	counter	Wh	Ex	REAL
Ex1	Export Energy 1	counter	Wh	Ex1	REAL
Ex2	Export Energy 2	counter	Wh	Ex2	REAL
Ex3	Export Energy 3	counter	Wh	Ex3	REAL
M	Mass	counter	g	M	REAL
PEC	People Count	counter		PEC	REAL
PIR	PIR Count Value	counter		PIR	DOUBLE PRECISION
PIRt	Time Since Last PIR Count	instant		PIR_t	DOUBLE PRECISION
RE	Reactive Energy	counter	VArh	RE	DOUBLE PRECISION
RE1	Reactive Energy 1	counter	VArh	RE1	REAL
RE2	Reactive Energy 2	counter	VArh	RE2	REAL
RE3	Reactive Energy 3	counter	VArh	RE3	REAL
REx	Reactive Energy Export	counter	VArh	REx	REAL
REx1	Reactive Energy Export 1	counter	VArh	REx1	REAL
REx2	Reactive Energy Export 2	counter	VArh	REx2	REAL
REx3	Reactive Energy Export 3	counter	VArh	REx3	REAL
Ton	On Time	counter	h	Ton	BIGINT
Top	Operating Time	counter	h	Top	BIGINT
Toff	Off Time	counter	h	Toff	BIGINT
Trun	Run Time	counter	h	Trun	BIGINT
Vol	Volume	counter	m^3	Vol	DOUBLE PRECISION
A12	Phase Angle V1 to V2	instant	degrees	A12	REAL
A13	Phase Angle V1 to V3	instant	degrees	A13	REAL
AC	Acceleration	instant	m/s	AC	REAL
AEr	Apparent Energy Relative	instant	VAh	AE_r	REAL
AEr1	Apparent Energy 1 Relative	instant	VAh	AE1_r	REAL
AEr2	Apparent Energy 2 Relative	instant	VAh	AE2_r	REAL
AEr3	Apparent Energy 3 Relative	instant	VAh	AE3_r	REAL
AF	Air Flow	instant	m/s	AF	REAL

AIN1	Analogue Input 1	instant	m/s	AI1	REAL
AIN2	Analogue Input 2	instant	m/s	AI2	REAL
AIN3	Analogue Input 3	instant	m/s	AI3	REAL
AIN4	Analogue Input 4	instant	m/s	AI4	REAL
AIN5	Analogue Input 5	instant	m/s	AI5	REAL
AIN6	Analogue Input 6	instant	m/s	AI6	REAL
AIN7	Analogue Input 7	instant	m/s	AI7	REAL
AIN8	Analogue Input 8	instant	m/s	AI8	REAL
AO1	Analogue Output 1	instant	m/s	AO1	REAL
AO2	Analogue Output 2	instant	m/s	AO2	REAL
AO3	Analogue Output 3	instant	m/s	AO3	REAL
AO4	Analogue Output 4	instant	m/s	AO4	REAL
AO5	Analogue Output 5	instant	m/s	AO5	REAL
AO6	Analogue Output 6	instant	m/s	AO6	REAL
AO7	Analogue Output 7	instant	m/s	AO7	REAL
AO8	Analogue Output 8	instant	m/s	AO8	REAL
AP	Atmospheric Pressure	instant	mBar	AP	REAL
BTUr	BTU Relative	instant		BTU_r	REAL
CO	Carbon Monoxide	instant	ppm	CO	REAL
CO2	Carbon Dioxide	instant	ppm	CO2	REAL
Cr	Pulse Count Relative	instant		C_r	REAL
D	Current THD	instant	%	D	REAL
D1	Current THD 1	instant	%	D1	REAL
D2	Current THD 2	instant	%	D2	REAL
D3	Current THD 3	instant	%	D3	REAL
Dmax	Current THD Max	instant	%	D_max	REAL
Dmax1	Current THD 1 Max	instant	%	D1_max	REAL
Dmax2	Current THD 2 Max	instant	%	D2_max	REAL
Dmax3	Current THD 3 Max	instant	%	D3_max	REAL
DICr1	Digital Input Counter 1 relative	instant		DI_C_r1	REAL
DICr2	Digital Input Counter 2 relative	instant		DI_C_r2	REAL

DICr3	Digital Input Counter 3 relative	instant		DI_C_r3	REAL
DICr4	Digital Input Counter 4 relative	instant		DI_C_r4	REAL
DICr5	Digital Input Counter 5 relative	instant		DI_C_r5	REAL
DICr6	Digital Input Counter 6 relative	instant		DI_C_r6	REAL
DICr7	Digital Input Counter 7 relative	instant		DI_C_r7	REAL
DICr8	Digital Input Counter 8 relative	instant		DI_C_r8	REAL
Er	Energy Relative	instant	Wh	E_r	REAL
Er1	Energy 1 Relative	instant	Wh	E1_r	REAL
Er2	Energy 2 Relative	instant	Wh	E2_r	REAL
Er3	Energy 3 Relative	instant	Wh	E3_r	REAL
Exr	Export Energy Relative	instant	Wh	Ex_r	REAL
Exr1	Export Energy 1 Relative	instant	Wh	Ex1_r	REAL
Exr2	Export Energy 2 Relative	instant	Wh	Ex2_r	REAL
Exr3	Export Energy 3 Relative	instant	Wh	Ex3_r	REAL
F	Frequency	instant	Hz	F	REAL
I	Current	instant	A	I	REAL
I1	Current 1	instant	A	I1	REAL
I2	Current 2	instant	A	I2	REAL
I3	Current 3	instant	A	I3	REAL
Imax	Current Max	instant	A	I_max	REAL
Imax1	Current 1 Max	instant	A	I1_max	REAL
Imax2	Current 2 Max	instant	A	I2_max	REAL
Imax3	Current 3 Max	instant	A	I3_max	REAL
In	Neutral Current	instant	A	In	DOUBLE PRECISION
Inmax	Neutral Current Max	instant	A	In_max	DOUBLE PRECISION
LL	Light Level	instant	lux	LL	DOUBLE PRECISION
Mf	Mass Flow	instant	g/h	Mf	REAL
Mfr	Mass Flow Relative	instant	g/h	Mf_r	REAL

NOX1	Nitrous Oxide 1	instant	ppm	NOX1	REAL
NOX2	Nitrous Oxide 2	instant	ppm	NOX2	REAL
P	Power	instant	W	P	REAL
P1	Power 1	instant	W	P1	REAL
P2	Power 2	instant	W	P2	REAL
P3	Power 3	instant	W	P3	REAL
Pmax	Power Max	instant	W	P_max	REAL
Pmax1	Power 1 Max	instant	W	P1_max	REAL
Pmax2	Power 2 Max	instant	W	P2_max	REAL
Pmax3	Power 3 Max	instant	W	P3_max	REAL
PE	Percentage	instant	%	PE	REAL
PECR	People Count Relative	instant		PECR	REAL
PF	Power Factor	instant		PF	REAL
PF1	Power Factor 1	instant		PF1	REAL
PF2	Power Factor 2	instant		PF2	REAL
PF3	Power Factor 3	instant		PF3	REAL
PFmax	Power Factor Max	instant		PF_max	REAL
PFmax1	Power Factor 1 Max	instant		PF1_max	REAL
PFmax2	Power Factor 2 Max	instant		PF2_max	REAL
PFmax3	Power Factor 3 Max	instant		PF3_max	REAL
PIRr	PIR Count Value Relative	instant		PIR_r	REAL
PM1	Particulate Matter 1	instant	ug/m^3	PM1	REAL
PM2	Particulate Matter 2	instant	ug/m^3	PM2	REAL
PM3	Particulate Matter 3	instant	ug/m^3	PM3	REAL
PM4	Particulate Matter 4	instant	ug/m^3	PM4	REAL
Pr	Pressure	instant	bar	Pr	REAL
Q	Reactive Power	instant	VAr	Q	REAL
Q1	Reactive Power 1	instant	VAr	Q1	REAL
Q2	Reactive Power 2	instant	VAr	Q2	REAL
Q3	Reactive Power 3	instant	VAr	Q3	REAL
Qmax	Reactive Power Max	instant	VAr	Q_max	REAL
Qmax1	Reactive Power 1 Max	instant	VAr	Q1_max	REAL

Qmax2	Reactive Power 2 Max	instant	VAr	Q2_max	REAL
Qmax3	Reactive Power 3 Max	instant	VAr	Q3_max	REAL
R	Rate	instant	/h	R	REAL
RAD	Radon	instant	Bq/m3	RAD	REAL
RES	Resistance	instant	Ohms	Res	REAL
REr	Reactive Energy Relative	instant	VArh	RE_r	REAL
REr1	Reactive Energy 1 Relative	instant	VArh	RE1_r	REAL
REr2	Reactive Energy 2 Relative	instant	VArh	RE2_r	REAL
REr3	Reactive Energy 3 Relative	instant	VArh	RE3_r	REAL
RExr	Reactive Energy Export Relative	instant	VArh	REx_r	REAL
RExr1	Reactive Energy Export 1 Relative	instant	VArh	REx1_r	REAL
RExr2	Reactive Energy Export 2 Relative	instant	VArh	REx2_r	REAL
RExr3	Reactive Energy Export 3 Relative	instant	VArh	REx3_r	REAL
RH	Relative Humidity	instant	%	RH	DOUBLE PRECISION
S	Apparent Power	instant	VA	S	REAL
S1	Apparent Power 1	instant	VA	S1	REAL
S2	Apparent Power 2	instant	VA	S2	REAL
S3	Apparent Power 3	instant	VA	S3	REAL
Smax	Apparent Power Max	instant	VA	S_max	REAL
Smax1	Apparent Power 1 Max	instant	VA	S1_max	REAL
Smax2	Apparent Power 2 Max	instant	VA	S2_max	REAL
Smax3	Apparent Power 3 Max	instant	VA	S3_max	REAL
SO	Sound	instant	dB	SO	REAL
T	Temperature	instant	^{o}C	T	REAL
T1	Temperature 1	instant	^{o}C	T1	REAL
T2	Temperature 2	instant	^{o}C	T2	REAL
T3	Temperature 3	instant	^{o}C	T3	REAL

T4	Temperature 4	instant	^oC	T4	REAL
T5	Temperature 5	instant	^oC	T5	REAL
T6	Temperature 6	instant	^oC	T6	REAL
T7	Temperature 7	instant	^oC	T7	REAL
T8	Temperature 8	instant	^oC	T8	REAL
Ti	Internal Temperature	instant	^oC	Ti	REAL
Te	External Temperature	instant	^oC	Te	REAL
Tf	Flow Temperature	instant	^oC	Tf	REAL
Tr	Return Temperature	instant	^oC	Tr	REAL
Td	Temperature Difference	instant	^oC	Td	REAL
U	Voltage (LL)	instant	V	U	REAL
U1	Voltage 1 (LL)	instant	V	U1	REAL
U2	Voltage 2 (LL)	instant	V	U2	REAL
U3	Voltage 3 (LL)	instant	V	U3	REAL
Umax	Voltage (LL) Max	instant	V	U_max	REAL
Umax1	Voltage 1 (LL) Max	instant	V	U1_max	REAL
Umax2	Voltage 2 (LL) Max	instant	V	U2_max	REAL
Umax3	Voltage 3 (LL) Max	instant	V	U3_max	REAL
V	Voltage (LN)	instant	V	V	DOUBLE PRECISION
V1	Voltage 1 (LN)	instant	V	V1	REAL
V2	Voltage 2 (LN)	instant	V	V2	REAL
V3	Voltage 3 (LN)	instant	V	V3	REAL
Vmax	Maximum System Voltage	instant	V	V_max	REAL
Vmax1	Voltage 1 (LN) Max	instant	V	V1_max	REAL
Vmax2	Voltage 2 (LN) Max	instant	V	V2_max	REAL
Vmax3	Voltage 3 (LN) Max	instant	V	V3_max	REAL
btryVoltage	Battery Voltage	instant	V	Vbat	DOUBLE PRECISION
Vf	Volume Flow	instant	m^3/h	Vf	REAL
Vfr	Volume Flow Relative	instant	m^3/h	Vf_r	REAL
VOC1	Volatile Organic Compounds 1	instant	ppb	VOC1	REAL

VOC2	Volatile Organic Compounds 2	instant	ppb	VOC2	REAL
Med	Medium	instant		Med	DOUBLE PRECISION
Stat	Status	instant		Stat	DOUBLE PRECISION