



CURVES INTEGRATION API

Users Guide

Version 1.10

[Summary](#)

A description of how to access data and services provided by the EcoGuard Curves Integration API.

Magnus Granberg
magnus.granberg@ecoguard.se

Index

Document history.....	4
Getting started	5
Security.....	5
Authentication.....	5
Encryption	5
API reference.....	5
Authentication.....	5
Authenticate.....	5
User	6
Read current user	6
Measuring Device	6
List measuring devices.....	6
Read measuring device	7
Manual Reading.....	8
List manual readings.....	8
Measuring Device Alert	9
List alerts	9
List alerts for measuring device.....	9
Node	9
List nodes.....	9
Read node.....	10
Node Type	11
List node types.....	11
Read node type	11
Measuring Point	12
List measuring points.....	12
List measuring points for node.....	13
Read measuring point	13
Installation.....	14
List installations	14
Comment.....	15
List comments	15

List comments for node.....	16
List comments for measuring point	16
Read comment	16
Create comment.....	17
Update comment	17
Delete comment.....	18
Data	18
Read data.....	18
Billing	20
List billings	20
Read billing result.....	21
Read transformed billing result.....	21
Reading.....	22
Import basic JSON.....	22
Import Elvaco.....	23
Rental Contract.....	23
Upload rental contract events.....	23
fastApi IMD reference	25
Consumption	26
List consumptions.....	26
Read consumption.....	26
Measurement.....	26
List measurements	26
Read measurement	26
Supply node	26
List supply nodes	26
Read supply node	26
Environments	27
Test environment	27
Production environment	27
Test	28
APPENDIX A – Standard billing XML format	29
APPENDIX B – Codes.....	33

Media code.....	33
Custom media codes can be added.....	33
Price component type	33
Rate unit	33
Quantity unit	33
Status codes for contract upload	33
APPENDIX C – Models.....	34
Domain membership.....	34
Measuring device register	34
Assembly	34
Listed node	34
Listed rental contract	34
Listed measuring point.....	34
Listed installation	35
Installation register	35
Property.....	35
Property value	35
Comment reference	35
Comment.....	35
Measuring device alert.....	36

Document history

Date	Version	Created by	Changes
2015-12-01	1.0	Magnus Granberg	
2016-04-14	1.1	Magnus Granberg	Added description of contract upload.
2016-04-20	1.2	Magnus Granberg	Updated description of JSON format for contract upload.
2016-10-13	1.3	Magnus Granberg	Updated status codes for contract upload
2017-01-09	1.4	Magnus Granberg	Added cURL-samples for authentication and contract upload.
2017-05-03	1.5	Magnus Granberg	Added Getting started and Authentication. Also, changed the way the API is documented and added information about fastApi IMD.
2020-02-07	1.6	Magnus Granberg	Added description of measuring devices, manual readings, nodes, measuring points and installations.
2020-05-18	1.7	Magnus Granberg	Added description of current user and node types.
2020-09-12	1.8	Magnus Granberg	Added description of data.
2020-10-02	1.9	Magnus Granberg	Added description of comments and device alerts.
2022-03-25	1.10	Magnus Granberg	Added examples for reading data.

Getting started

EcoGuards Curves is a cloud-based SaaS solution that collects, processes, and analyzes sensor data. Developers can access data and services to utilize in their own apps or to enhance Curves functionality. This document describes how to access data and services with an API provided by EcoGuard.

Curves Integration API has support for fastAPI IMD, specified by SABO. For more information about fastAPI IMD, see <http://www.fastapi.se/apidocimd/v1/>.

Security

Authentication

Most requests to Curves Integration API require authentication. Hence, each request to the API must provide a valid access token. An access token is unique to a user and includes information about when the token will expire.

To obtain an access token, the caller:

1. Encodes its credentials (username and password)
2. Makes a request to the POST /api/token endpoint to exchange these credentials for an access token
3. When accessing the API, the caller uses the access token to authenticate

Integration accounts with username and password are created in Curves.

Encryption

All requests to the API must be made over HTTPS.

API reference

Authentication

Authenticate

Authenticates user credentials and issues an access token to be used in subsequent calls to the API.

POST /api/token

Request

Data

grant_type	string	Always the value password	Required
username	string	Obtained user name	Required
password	string	Obtained password	Required

CURL-sample

```
curl -i --X POST "https://integration-test.ecoguard.se/Token H "Content-Type: application/x-www-form-urlencoded" -d "grant_type=password&username=your_username&password=your_password"
```

Response

Access token as string.

Example:

Access-Token: mF_9.B5f-4.1JqM

User

Read current user

Reads information about current user, that is, user for which access token was created.

GET /api/users/self

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

Single user or empty.

givenName	string	Given name for user.	Optional
SurName	string	Sur name for user.	Optional
domains	array [Domain membership]	Domains that user is a member of.	Optional

Measuring Device

List measuring devices

Lists measuring devices in a domain.

GET /api/**domainCode**/measuringdevices

Parameters

domainCode	string	Code for the domain which measuring devices should be listed.	Required
status	array [string]	Current status of the measuring device. Valid values are: <ul style="list-style-type: none"> active delivery project disposed 	Optional
externalKey	string	External, or public, key of the measuring device. That can, for instance, be a serial number. The external key is not guaranteed to be unique.	Optional

internalKey	string	Unique key of the measuring device.	Optional
-------------	--------	-------------------------------------	----------

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

List of measuring devices

id	int	Identity of measuring device.	Required
externalKey	string	External, or public, key of the measuring device.	Required
internalKey	string	Unique key of the measuring device.	Required
assemblyType	string		Required
deviceType	string		Required
status	string	Current status of the measuring device.	Required
domainCode	string	Code for the domain which measuring device belongs to.	Required
currentMeasuringPointID	int	Identity of the measuring point where the measuring device is installed.	Optional
properties	array [Property Value]	Properties for the measuring device.	Optional

Read measuring device

Reads a single measuring device in a domain.

GET /api/**domainCode**/measuringdevices/**measuringDeviceID**

Parameters

domainCode	string	Code for the domain which measuring devices should be listed.	Required
measuringDeviceID	int	Identity of the measuring device.	Required

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

Measuring device.

id	int	Identity of measuring device.	Required
externalKey	string	External, or public, key of the measuring device.	Required

internalKey	string	Unique key of the measuring device.	Required
assemblyType	string		Required
deviceType	string		Required
status	string	Current status of the measuring device.	Required
domainCode	string	Code for the domain which measuring device belongs to.	Required
currentMeasuringPointID	int	Identity of the measuring point where the measuring device is installed.	Optional
properties	array [Property Value]	Properties for the measuring device.	Optional
registers	array [Measuring device register]	Registers for the measuring device.	Optional

Manual Reading

List manual readings

Lists manually registered readings for a measuring device. Readings are registered per register since a measuring device can have multiple registers. Hence, manual readings are read by register.

GET `/api/domainCode/measuringdevices/measuringDeviceID/registers/registerNumber/readings`

Parameters

domainCode	string	Code for the domain which measuring devices should be listed.	Required
measuringDeviceID	int	Identity of the measuring device.	Required
registerNumber	int	Number of the register which readings should be listed.	Required

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

List of manual readings.

id	int	Identity of the manual reading.	Required
time	datetime	Timestamp for the manual reading.	Required
value	double	Value of manual reading.	Required
readBy	string	Information about the person who made the manual reading.	Optional

Measuring Device Alert

List alerts

Lists alerts in the domain.

GET /api/**domainCode**/devicealerts

Parameters

domainCode	string	Code for the domain which comments should be listed.	Required
measuringDeviceID	Array [int]	List of identities of measuring devices.	Required

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

List of Measuring device alert or empty.

List alerts for measuring device

Lists alerts for a measuring device in the domain.

GET /api/**domainCode**/measuringdevices/**measuringDeviceID**/devicealerts

Parameters

domainCode	string	Code for the domain which comments should be listed.	Required
measuringDeviceID	int	Identities of measuring device.	Required

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

Single Measuring device alert or empty.

Node

List nodes

Lists nodes in the domain.

GET /api/**domainCode**/nodes

Parameters

domainCode	string	Code for the domain which nodes should be listed.	Required
------------	--------	---	----------

nodeID	int	Identity of the node to be listed.	Optional
nodeType	int	Type of the nodes to be listed.	Optional
name	string	Name of nodes to be listed.	Optional
objectNumber	string	Object number for the nodes to be listed.	Optional
includeSubNodes	bool	Specifies if sub-nodes for the matched nodes also should be listed. By default, no sub-nodes are read.	Optional

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

List of nodes.

id	int	Identity of node.	Required
nodeType	int	Type of the node.	Optional
name	string	Name of node.	Required
objectNumber	string	Object number for the node.	Optional
parentNodeID	int	Identity of parent node. All nodes except the root node in the domain has a parent node.	Optional

Read node

Reads a single node in a domain.

GET /api/***domainCode***/nodes/***nodeID***

Parameters

domainCode	string	Code for the domain which node should be read.	Required
nodeID	int	Identity of the node.	Required

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

Single node or empty.

id	int	Identity of node.	Required
nodeType	int	Type of the node.	Optional
name	string	Name of node.	Required
objectNumber	string	Object number for the node.	Optional

parentNodeID	int	Identity of parent node. All nodes except the root node in the domain has a parent node.	Optional
properties	array [Property Value]	Properties for the node.	Optional
measuringPoints	array [Listed measuring point]	Measuring points located directly under the node.	Optional
rentalContracts	array [Listed rental contract]	Rental contracts for the node.	Optional
subNodes	array [Listed node]	Sub-nodes located directly under the node.	Optional

Node Type

List node types

Lists node types in the domain.

GET /api/**domainCode**/nodetypes

Parameters

domainCode	string	Code for the domain which nodes should be listed.	Required
------------	--------	---	----------

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

List of node types.

id	int	Identity of node type.	Required
code	int	Unique code for the node type.	Optional
name	string	Name of node type.	Required

Read node type

Reads a single node type in a domain.

GET /api/**domainCode**/nodetypes/**nodeTypeID**

Parameters

domainCode	string	Code for the domain which node types should be read.	Required
nodeTypeID	int	Identity of the node type.	Required

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

Single node type or empty.

id	int	Identity of node type.	Required
code	int	Unique code for the node type.	Optional
name	string	Name of node type.	Required
isRentable	bool	Specifies if nodes of this type are rentable or not, that is, can have an object number, can be included in billing, and used in Insight.	Required
parentNodeTypeID	int	Identity of the parent node type, if any. For example, node type Room can be parent for node type Kitchen.	Optional
properties	array [Property]	Properties for the node type.	Optional

Measuring Point

List measuring points

Lists measuring points in the domain.

GET `/api/domainCode/measuringpoints`

Parameters

domainCode	string	Code for the domain which nodes should be listed.	Required
nodeID	int	Identity of the node which measuring points should be listed.	Optional
includeSubNodes	bool	Specifies if measuring points for sub-nodes also should be listed. Only applicable if nodeID is specified.	Optional
name	string	Name of measuring points to be listed.	Optional
facilityID	string	Facility ID of measuring points to be listed.	Optional

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

List of measuring points.

id	string	Identity of the measuring point.	Required
nodeID	int	Identity of the node which measuring points should be listed.	Optional
name	string	Name of measuring point.	Required
facilityID	string	Facility ID of measuring point.	Optional

nodeID	int	Identity of the node under which the measuring point is located.	Required
--------	-----	--	----------

List measuring points for node

Lists measuring points for a node in the domain.

GET /api/**domainCode**/nodes/**nodeID**/measuringpoints

Parameters

domainCode	string	Code for the domain which nodes should be listed.	Required
nodeID	int	Identity of the node which measuring points should be listed.	Required
includeSubNodes	bool	Specifies if measuring points for sub-nodes also should be listed. By default, no sub-nodes are read.	Optional
name	string	Name of measuring points to be listed.	Required
facilityID	string	Facility ID of measuring points to be listed.	Optional

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

List of measuring points.

id	string	Identity of the measuring point.	Required
nodeID	int	Identity of the node which measuring points should be listed.	Optional
name	string	Name of measuring point.	Required
facilityID	string	Facility ID of measuring point.	Optional
nodeID	int	Identity of the node under which the measuring point is located.	Required

Read measuring point

Reads a single measuring point in a domain.

GET /api/**domainCode**/measuringpoints/**measuringPointID**

Parameters

domainCode	string	Code for the domain which node should be read.	Required
------------	--------	--	----------

measuringPointID	int	Identity of the measuring point.	Required
------------------	-----	----------------------------------	----------

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

Single measuring point or empty.

id	string	Identity of the measuring point.	Required
nodeID	int	Identity of the node which measuring points should be listed.	Optional
name	string	Name of measuring point.	Required
facilityID	string	Facility ID of measuring point.	Optional
nodeID	int	Identity of the node under which the measuring point is located.	Required
installations	array [Listed installation]	Installations at the measuring point.	Optional
nodes	array [Listed node]	Hierarchy for nodes under which the measuring point is located.	Required

Installation

List installations

Lists installations in the domain. An installation is a measuring device linked to a measuring point.

GET /api/**domainCode**/installations

Parameters

domainCode	string	Code for the domain which installations should be listed.	Required
nodeID	int	Identity of the node for which installations should be listed.	Optional
measuringPointID	int	Identity of the measuring point which installations should be listed.	Optional
startDate	datetime	Minimum start date for installation.	Optional
endDate	datetime	Maximum end date for installation.	Optional

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

List of installations.

domainCode	string	Code for the domain to which the installation belongs.	Required
nodeID	int	Identity of the node under which the measuring point is located.	Required
measuringPointID	int	Identity of the measuring point for the installation.	Required
startDate	datetime	Start date for installation.	Required
endDate	datetime	End date for installation.	Optional
registers	array [Installation register]	Description of how measuring device registers should be used at the measuring point.	Optional

Comment

List comments

Lists comments in the domain.

GET /api/**domainCode**/comments

Parameters

domainCode	string	Code for the domain which comments should be listed.	Required
priority	Array [byte]	Priority of the comments to be read. Priorities:	Optional
from	long	Comment time in Unix time, seconds since January 1 1970.	Optional
to	long	Comment time in Unix time, seconds since January 1 1970.	Optional
createdFrom	long	Comment creation time in Unix time, seconds since January 1 1970.	Optional
createdTo	long	Comment creation time in Unix time, seconds since January 1 1970.	Optional
text	string	Comment text.	Optional
userID	string	Identity (guid) of the user who created the comment.	Optional

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

List of Comment or empty.

List comments for node

Lists comments for a node in the domain.

GET /api/**domainCode**/nodes/**nodeID**/comments/

Parameters

domainCode	string	Code for the domain which node should be read.	Required
nodeID	int	Identity of the node.	Required

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

List of Comment or empty.

List comments for measuring point

Lists comments for a measuring point in the domain.

GET /api/**domainCode**/measuringpoints/**measuringPointID**/comments/

Parameters

domainCode	string	Code for the domain which node should be read.	Required
measuringPointID	int	Identity of the measuring point.	Required

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Response

List of Comment or empty.

Read comment

Reads a single comment in a domain.

GET /api/**domainCode**/comments/**commentID**

Parameters

domainCode	string	Code for the domain which node should be read.	Required
commentID	int	Identity of the comment.	Required

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Response

Single Comment or empty.

Create comment

Creates a single comment in a domain.

POST /api/***domainCode***/comments/

Parameters

domainCode	string	Code for the domain which node should be created.	Required
------------	--------	---	----------

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Data

text	string	Comment text.	Optional
commentDate	long	Comment time in Unix time, seconds since January 1 1970.	Optional
priority	byte	Priority of the comment.	Required
refersTo	Comment reference	Type and identity of the object that the comment refers to.	Required

Response

Single Comment or empty.

Update comment

Updates a single comment in a domain.

PUT /api/***domainCode***/comments/***commentID***

Parameters

domainCode	string	Code for the domain which comment should be updated.	Required
commentID	int	Identity of the comment.	Required

Request

Headers

Authentication	Bearer <i>access_token</i>
X-Version	Version of requested API

Data

text	string	Comment text.	Optional
------	--------	---------------	----------

commentDate	long	Comment time in Unix time, seconds since January 1 1970.	Optional
priority	byte	Priority of the comment.	Required

Response

Single Comment or empty.

Delete comment

Deletes a single comment in a domain.

DELETE /api/**domainCode**/comments/**commentID**

Parameters

domainCode	string	Code for the domain which comment should be updated.	Required
commentID	int	Identity of the comment.	Required

Request

Headers

Authentication	Bearer access_token
X-Version	Version of requested API

Data

Read data

Reads data, by measuring device or aggregated.

GET /api/**domainCode**/data

Parameters

domainCode	string	Code for the domain which data should be read.	Required
One of the five following selections must be specified:			
groupName	string	Name of group (collection of nodes and measuring points)	Optional
nodeID	int	Identity of node (city, estate, apartment etc)	Optional
measuringPointID	int	Identity of measuring point	Optional
devicePublicKey	string	Public key for measuring device	Optional
deviceID	int	Identity of measuring device (meter, sensor etc.)	Optional
includeSubNodes	boolean	Specifies if sub-nodes should be included. Only applicable if nodeID has been specified. Default value is false.	Optional

from	integer	Start of period in Unix time, seconds since January 1 1970.	Optional
to	integer	End of period in Unix time, seconds since January 1 1970. Default values is now.	Optional
interval	integer	Interval to be used by series functions: <ul style="list-style-type: none"> • Q = Quarter • H = Hour • D = Day • M = Month 	Required
grouping	string	Type of object to group on: <ul style="list-style-type: none"> • device • measuringPoint • apartment • nodeType:ID <p>For nodeType ID is the identity of a node type in the domain. It can be the ID of a standard node type such building, but it can also be a customer defined node type.</p> <p>If no grouping is specified, both single values and series, will be returned per device.</p>	Optional
utl	array [string]	Array of one or more utility codes. One or more functions must be specified per utility code. <p>Unit per function is optional. If no unit is specified, the default unit for the utility will be used. For example, kWh is used for electricity and °C for temperature.</p> <p>Syntax: utl[func 1:unit, func 2:unit, ..., func n:unit]</p>	Required

Functions

Functions that can be used when reading data.

Name	Series/ Single value	Description
val	Series	Series with values per device and interval (meter reading for consumption meters). Cannot be used with grouping. In future release, grouping will be ignored.
avg	Series	Series with average value per interval
max	Series	Series with maximum value per interval
min	Series	Series with minimum value per interval
med	Series	Series with median value per interval
con	Series	Series with consumption per interval
lastval	Single value	Last value per device for the specified period (meter reading for consumption meters). Cannot be used with grouping. In future release, grouping will be ignored.
totavg	Single value	Total average for the specified period
absmax	Single value	Absolute maximum for the specified period
absmin	Single value	Absolute minimum for the specified period
sumcon	Single value	Total consumption for the specified period

Samples

Get electricity (media code ELEC) reading per hour for meter with public serial number 90011764 during period January 4 2020 23:00:00 to January 10 2020 23:00:00

`https://integration-test.ecoguard.se/api/DEMO/data?
devicePublicKey=90011764&from=1578178800&to=1578697200&interval=H&utl=ELEC[val]`

Get average and minimum temperature per day for all apartments under the node with ID 150, during the period January 4 2020 23:00:00 to January 10 2020 23:00:00

`https://integration-test.ecoguard.se/api/DEMO/data?nodeID=150&includeSubNodes=true
&from=1578178800&to=1578697200&interval=D&grouping=apartment&utl=T[avg,min]`

Get hot water (media code HW) consumption per month for all nodes with type ID 20 under the node with ID 280, during the period January 4 2020 23:00:00 to January 10 2020 23:00:00

`https://integration-test.ecoguard.se/api/DEMO/data?
nodeID=280&from=1578178800&to=1578697200&interval=M&grouping=nodeType:20&utl=HW[con
]`

Billing

List billings

Lists all approved, none discarded billings for the specified domain.

GET /api/domainCode/billings

Parameters

domainCode	string	Code for the domain which billings should be listed.
------------	--------	--

Request

Headers

Authentication	Bearer <i>access_token</i>
----------------	-----------------------------------

Response

JSON formatted billings.

Read billing result

Reads a specified billing result.

GET /api/*domainCode*/billings/*billingID*

Parameters

domainCode	string	Code for the domain that the billing belongs to.
billingID	int	Identity of the billing that should be retrieved.

Request

Headers

Authentication	Bearer <i>access_token</i>
----------------	-----------------------------------

Response

JSON formatted billing result.

Read transformed billing result

Reads the specified billing result, transformed to the specified export format.

GET /api/*domainCode*/billings/*billingID*/xml/*exportFormat*

Parameters

domainCode	string	Code for the domain that the billing belongs to.
billingID	int	Identity of the billing that should be retrieved.
exportFormat	string	Name of the transformation that should be used.

Request

Headers

Authentication	Bearer <i>access_token</i>
----------------	-----------------------------------

Response

Billing transformed to the specified export format. The export format can i.e. be XML or CSV. If exportFormat **EcoGuard** is specified, the billing is transformed to EcoGuard's standard XML format (see APPENDIX A – Standard billing XML format).

Reading

Readings can be imported in various formats.

Each sensor must have a unique identity. If the system importing readings doesn't know the sensors identity, a unique identity can instead be built by combining other properties such as object number, media etc.

Reading belongs to the sensor that registered the reading. Curves has support for storing readings for multiple data records, i.e. both temperature and humidity, for a sensor.

Both sensors and data records must be registered before readings can be imported. Unknown sensors are logged and can later be viewed and activated in Curves.

Import basic JSON

Imports readings in a basic JSON format.

Request

Data

readings	JSON formatted list	List of readings
----------	---------------------	------------------

```
"$schema": "http://json-schema.org/draft-04/schema#",
"definitions": {},
"id": "http://ecoguard.se/reading-basic.json",
"properties": {
  "sensors": {
    "items": {
      "properties": {
        "datarecords": {
          "items": {
            "properties": {
              "number": {
                "type": "integer"
              }
            },
            "values": {
              "items": {
                "properties": {
                  "timestamp": {
                    "type": "string"
                  },
                  "value": {
                    "type": "string"
                  }
                }
              },
              "required": [
                "timestamp",
                "value"
              ],
              "type": "object"
            },
            "type": "array"
          }
        },
        "required": [
          "values"
        ]
      }
    }
  }
}
```

```

    "number"
  ],
  "type": "object"
},
"array": "array"
},
"serialnumber": {
  "type": "string"
}
},
"required": [
  "serialnumber",
  "datarecords"
],
"array": "array"
},
"required": [
  "sensors"
],
"array": "array"
}

```

Response

0 if import was successful and 1 if import failed.

Import Elvaco

Imports readings in a format that is used by Elvaco CMe devices. The format is described in the document "Report Templates Description Template 3101" that can be downloaded from <http://www.elvaco.com/sv/download/1538>.

Response

0 if import was successful and 1 if import failed.

Rental Contract

Upload rental contract events

Uploads rental contract events for the specified domain. A contract event is a creation of a new contract or a change, termination or discard of an existing contract.

POST /api/**domainCode**/contracts/

Parameters

domainCode	string	Code for the domain which contract events should be uploaded.
------------	--------	---

Request

Headers

Authentication	Bearer access_token
----------------	----------------------------

Data

contractEvents	JSON formatted list	List of contract events
----------------	---------------------	-------------------------


```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "ObjectNumber": {
        "type": "string"
      },
      "ContractCode": {
        "type": "string"
      },
      "Validity": {
        "type": "object",
        "properties": {
          "Start": {
            "type": "string",
            "description": "As date-time string, i.e. \"2015-01-01T00:00:00\""
          },
          "Termination": {
            "type": "string",
            "description": "As date-time string, i.e. \"2015-02-01T00:00:00\""
          }
        }
      },
      "Discarded": {
        "type": "boolean"
      }
    },
    "required": [
      "ObjectNumber",
      "ContractCode",
      "Validity"
    ]
  },
  "required": [
    "0"
  ]
}
```

CURL-sample

```
curl -i -X POST https://integration-test.ecoguard.se/api/DEMO/contracts -H "Content-Type: application/json" -H "Authorization: Bearer oj_dCzFOXLHbcawil6ts4WeRRtka-dHFHdOdClGSRVaaw08eCtyB1EDPpk5v4Gaf2n8-BpkT07M7pgbDcWktcntCwwOtLqDpqhZfhZg66CQNatO7rbTXOnZ6a56IGUos0D1oRiX07mKKTDURD9Lo0kT012SteZaaBLt3xQ3Vs8b_PTZfBQITyByPTyV27KVXRJffDIJiQ8CILSx3r3-AR-YbDfTsIT3EOdXuc30O8h7o_o06JMIb_EhoocuLZyLNS4KPYzMDLzT1uDpKHng5YI2LBM3BixIP_1aqIDv-IJXgOuxBXijE3DKliHGh946mz91R2cDfj0trD92CesOw7h171T78LEtAhRh2umNZW0JGoQbfnhsCin5symRi4i6jQM9ymVQpM0hPYGpDDJVA" -d "[{"ObjectNumber\":\"3010-1101\", \"ContractCode\":\"3010-1101-02\", \"Validity\":{\"Start\":\"2015-01-01T00:00:00\", \"Termination\":null}, \"Discarded\":false}]"
```

Response

```
{
  "type": "array",
```

```
"items": {
  "type": "object",
  "properties": {
    "ObjectNumber": {
      "type": "string"
    },
    "ContractCode": {
      "type": "string"
    },
    "Validity": {
      "type": "object",
      "properties": {
        "Start": {
          "type": "string",
          "description": "As date-time string, i.e. \"2015-01-01T00:00:00\""
        },
        "Termination": {
          "type": "string",
          "description": "As date-time string, i.e. \"2015-02-01T00:00:00\""
        }
      }
    },
    "Discarded": {
      "type": "boolean"
    },
    "Status": {
      "type": "object",
      "properties": {
        "StatusCode": {
          "type": "string"
        },
        "StatusText": {
          "type": "string"
        }
      }
    }
  },
  "required": [
    "ObjectNumber",
    "ContractCode",
    "Validity",
    "Status"
  ]
},
"required": [
  "0"
]
}
```

For a description of status codes, see APPENDIX B – Codes.

fastApi IMD reference

In this chapter, only http verbs and urls are described. For a complete fastApi IMD reference, see <http://www.fastapi.se/apidocimd/v1/ApiDatamodel/api-datamodel>.

The resources debit and debit price are not implemented.

Consumption

List consumptions

GET /fastapi/*domainCode*/v1/api/fi2mediaconsumption

Read consumption

GET /fastapi/*domainCode*/v1/api/fi2mediaconsumption/*id*

Measurement

List measurements

GET /fastapi/*domainCode*/v1/api/fi2measurement

Read measurement

GET /fastapi/*domainCode*/v1/api/fi2measurement/*id*

Supply node

List supply nodes

GET /fastapi/*domainCode*/v1/api/fi2supplynode

Read supply node

GET /fastapi/*domainCode*/v1/api/fi2supplynode/*id*

Environments

Test environment

<https://integration-test.ecoguard.se>

Production environment

<https://integration.ecoguard.se>

Test

For both test and production environment, the API can be tested from a web page. For each method in the API, there is a form that can be used to call the method. The response from each method is presented at the bottom of the page under the heading Response.

CURVES Integration

Login

User ID

Password

Login

Logout

Billings

Domain

Get all billings

Billing

Domain

Billing ID

APPENDIX A – Standard billing XML format

EcoGuard's standard XML format.

- Billing (1..1)
 - ID [(1..1)]
 - Name (1..1)
 - Created (1..1)
 - BillingResult (0..*)
 - ObjectNumber (1..1)
 - Contract (0..*)
 - ContractResultID (1..1)
 - ContractCode (0..1)
 - Start (1..1)
 - End (1..1)
 - BillingMedia (0..*)
 - MediaCode (1..1)
 - BillingItem (0..*)
 - MediaCode (1..1)
 - PriceListName (1..1)
 - PriceComponentName (1..1)
 - PriceComponentType (1..1)
 - Start (1..1)
 - End (1..1)
 - Quantity (1..1)
 - QuantityUnit (1..1)
 - Rate (1..1)
 - RateUnit (1..1)
 - Total (1..1)
 - SerialNumber (0..1)
 - ReadingStart (1..1)
 - ReadingEnd (0..1)
 - VAT (1..1)
 - Rounding (1..1)
 - BillingAlert (0..*)
 - Severity (1..1)
 - Code (1..1)
 - Text (1..1)
 - MeterSummary (0..*)
 - MediaCode (1..1)
 - Start (1..1)
 - End (1..1)
 - Quantity (1..1)
 - QuantityUnit (1..1)
 - TotalConsumed (1..1)

- TotalFixed (1..1)
- TotalIncluded (1..1)
- SerialNumber (1..1)
- ReadingStart (1..1)
- ReadingEnd (1..1)

Exempel

```
<?xml version="1.0"?>
<Billing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ID>2002</ID>
  <Name>Debitering 2015-11-25 09:00</Name>
  <Created>2015-11-25T09:00:43.593</Created>
  <BillingResults>
    <BillingResult>
      <ObjectNumber>5103-101-1-103</ObjectNumber>
      <Contracts>
        <Contract>
          <ContractResultID>1028</ContractResultID>
          <Start>2015-01-01T00:00:00</Start>
          <End>2015-02-01T00:00:00</End>
          <BillingMedia>
            <BillingMedia>
              <MediaCode>ELEC</MediaCode>
              <BillingItems>
                <BillingItem>
                  <MediaCode>ELEC</MediaCode>
                  <PriceListName>Prislista el</PriceListName>
                  <PriceComponentName>Fast avgift, nät</PriceComponentName>
                  <PriceComponentType>T1</PriceComponentType>
                  <Start>2015-01-01T00:00:00</Start>
                  <End>2015-02-01T00:00:00</End>
                  <Quantity>31</Quantity>
                  <QuantityUnit>day</QuantityUnit>
                  <Rate>400</Rate>
                  <RateUnit>year</RateUnit>
                  <Total>33.97</Total>
                  <PositionName />
                  <ReadingStart>0</ReadingStart>
                  <ReadingEnd>0</ReadingEnd>
                </BillingItem>
                <BillingItem>
                  <MediaCode>ELEC</MediaCode>
                  <PriceListName>Prislista el</PriceListName>
                  <PriceComponentName>Förbrukning, el</PriceComponentName>
                  <PriceComponentType>C1</PriceComponentType>
                  <Start>2015-01-10T00:00:00</Start>
                  <End>2015-02-01T00:00:00</End>
                  <Quantity>199</Quantity>
                  <QuantityUnit>kWh</QuantityUnit>
                  <Rate>0.4</Rate>
                  <RateUnit>kWh</RateUnit>
                  <Total>79.6</Total>
                  <PositionName />
                  <SerialNumber>092700213556</SerialNumber>
                  <ReadingStart>15252</ReadingStart>
                  <ReadingEnd>15451</ReadingEnd>
                </BillingItem>
                <BillingItem>
                  <MediaCode>ELEC</MediaCode>
                  <PriceListName>Prislista el</PriceListName>
                  <PriceComponentName>Energiskatt</PriceComponentName>
                  <PriceComponentType>C1</PriceComponentType>
                  <Start>2015-01-10T00:00:00</Start>
                  <End>2015-02-01T00:00:00</End>
                  <Quantity>0.199</Quantity>
                  <QuantityUnit>MWh</QuantityUnit>
                  <Rate>250</Rate>
                  <RateUnit>MWh</RateUnit>
                  <Total>49.75</Total>
                  <PositionName />
                </BillingItem>
              </BillingItems>
            </BillingMedia>
          </Contract>
        </Contracts>
      </BillingResult>
    </BillingResults>
  </Billing>
```

```

<SerialNumber>092700213556</SerialNumber>
<ReadingStart>15.252</ReadingStart>
<ReadingEnd>15.451</ReadingEnd>
</BillingItem>
<BillingItem>
  <MediaCode>ELEC</MediaCode>
  <PriceListName>Prislista el</PriceListName>
  <PriceComponentName>Inkluderat i hyra, elenergi</PriceComponentName>
  <PriceComponentType>D1</PriceComponentType>
  <Start>2015-01-01T00:00:00</Start>
  <End>2015-02-01T00:00:00</End>
  <Quantity>34</Quantity>
  <QuantityUnit>kWh</QuantityUnit>
  <Rate>0.65</Rate>
  <RateUnit>kWh</RateUnit>
  <Total>-22.1</Total>
  <PositionName />
  <ReadingStart>0</ReadingStart>
  <ReadingEnd>0</ReadingEnd>
</BillingItem>
</BillingItems>
<VAT>35.31</VAT>
<Rounding>0.47</Rounding>
</BillingMedia>
<BillingMedia>
  <MediaCode>HW</MediaCode>
  <BillingItems>
    <BillingItem>
      <MediaCode>HW</MediaCode>
      <PriceListName>Prislista varmvatten A</PriceListName>
      <PriceComponentName>Rörlig avgift, varmvatten</PriceComponentName>
      <PriceComponentType>C1</PriceComponentType>
      <Start>2015-01-01T00:00:00</Start>
      <End>2015-02-01T00:00:00</End>
      <Quantity>1.215</Quantity>
      <QuantityUnit>m3</QuantityUnit>
      <Rate>50</Rate>
      <RateUnit>m3</RateUnit>
      <Total>60.75</Total>
      <PositionName />
      <SerialNumber>97704047</SerialNumber>
      <ReadingStart>92.655</ReadingStart>
      <ReadingEnd>93.87</ReadingEnd>
    </BillingItem>
  </BillingItems>
  <VAT>15.19</VAT>
  <Rounding>0.06</Rounding>
</BillingMedia>
</BillingAlerts>
<BillingAlert>
  <Severity>1</Severity>
  <Code>200</Code>
  <Text>Värdet 141,22 kr för elenergi är mindre än rimlighetsvärdet 300 kr</Text>
</BillingAlert>
<BillingAlert>
  <Severity>0</Severity>
  <Code>0</Code>
  <Text>Inga rimlighetsvärden hittade för varmvatten</Text>
</BillingAlert>
</BillingAlerts>
<Summary>
  <MeterSummary>
    <MediaCode>ELEC</MediaCode>
    <Start>2015-01-01T00:00:00</Start>
    <End>2015-02-01T00:00:00</End>
    <Quantity>199</Quantity>
    <QuantityUnit>kWh</QuantityUnit>
    <TotalConsumed>129.35</TotalConsumed>
    <TotalFixed>33.97</TotalFixed>
    <TotalIncluded>-22.1</TotalIncluded>
    <SerialNumber>092700213556</SerialNumber>
    <ReadingStart>15252</ReadingStart>
    <ReadingEnd>15451</ReadingEnd>
  </MeterSummary>

```



```
<MeterSummary>
  <MediaCode>HW</MediaCode>
  <Start>2015-01-01T00:00:00</Start>
  <End>2015-02-01T00:00:00</End>
  <Quantity>1.215</Quantity>
  <QuantityUnit>m3</QuantityUnit>
  <TotalConsumed>60.75</TotalConsumed>
  <TotalFixed>0</TotalFixed>
  <TotalIncluded>0</TotalIncluded>
  <SerialNumber>97704047</SerialNumber>
  <ReadingStart>92.655</ReadingStart>
  <ReadingEnd>93.87</ReadingEnd>
</MeterSummary>
</Summary>
</Contract>
</Contracts>
</BillingResult>
</BillingResults>
</Billing>
```

APPENDIX B – Codes

Media code

Code	Name
ELEC	Electricity
HEAT	Heating
HW	Hot water
CW	Cold water
W ¹	Water
T	Temperature

Custom media codes can be added.

Price component type

Code	Description
T1	Fixed fee
C1	Fee for a consumed unit, i.e. the fee for a kWh
C2	Fee for consumed temperature
D1	Consumption included in rental
D2	Consumption included in rental, based on number of rooms
E1	Fee for an effect top

Rate unit

Code	Name
year	Year
kWh	kWh
MWh	MWh
m3	m ³

Quantity unit

Code	Name
day	Day
kWh	kWh
MWh	MWh
m3	m ³

Status codes for contract upload

Code	Description
0	Contract event was uploaded without error
1	Unknown object number
2	Unknown contract code
3	Unknown error when handling contract event

¹ Used for BillingMedia when "Merge charge for hot water and cold water" is checked in billing settings

4	Failed to create contract
5	Failed to update contract
6	Failed to terminate contract
7	Failed to discard contract
8	Invalid start or termination date
9	Invalid object number
10	Invalid contract code
11	Start or termination date overlap existing contract
12	Validity is not set

APPENDIX C – Models

Domain membership

code	int	Code for the domain.	Required
name	string	Name of the domain.	Required

Measuring device register

number	int	Register number.	Required
interval	string	Interval for register values.	Required
valueType	string	Type of value for register.	Required
quantity	string	Quantity for register.	Required
properties	array [property]	Properties for the register.	Optional

Assembly

date	datetime	Date and time for the assembly, that is, when an external measuring device was assembled to the host measuring device.	Required
hostDeviceID	int	Identity of the host measuring device.	Required
externalDeviceID	int	Identity of the external measuring device.	Required

Listed node

id	int	Identity of the node.	Optional
name	string	Name of the property.	Optional
value	string	Value of the property.	Optional

Listed rental contract

id	int	Identity of the rental contract.	Required
date	datetime	Start date for the rental contract.	Required
contractCode	string	Code for the rental contract.	Optional

Listed measuring point

id	string	Identity of the measuring point.	Required
----	--------	----------------------------------	----------

nodeID	int	Identity of the node which measuring points should be listed.	Optional
name	string	Name of measuring point.	Required
facilityID	string	Facility ID of measuring point.	Optional
nodeID	int	Identity of the node under which the measuring point is located.	Required

Listed installation

domainCode	string	Code for the domain to which the installation belongs.	Required
nodeID	int	Identity of the node under which the measuring point is located.	Required
measuringPointID	int	Identity of the measuring point for the installation.	Required
startDate	datetime	Start date for installation.	Required
endDate	datetime	End date for installation.	Optional
registers	array [installation register]		Optional

Installation register

register	int	Register number	Required
utilityCode	string	Utility that describes how the registered value should be used.	Required

Property

id	int	Identity of the property.	Required
code	int	Code for the property.	Required
name	string	Name of the property.	Optional

Property value

code	int	Code for the property.	Optional
name	string	Name of the property.	Optional
value	string	Value of the property.	Optional

Comment reference

type	int	Type of object that the comment refers to: 1. 0 = Unknown 2. 7 = Node 3. 11 = Measuring point	Required
id	int	Identity of the object that the comment refers to.	Required

Comment

id	int	Identity of comment.	Required
----	-----	----------------------	----------

text	string	Comment text.	Optional
commentDate	long	Comment time in Unix time, seconds since January 1 1970.	Optional
priority	byte	Priority of the comment.	Required
refersTo	Comment reference	Type and identity of the object that the comment refers to.	Required
created	long	Comment creation time in Unix time, seconds since January 1 1970.	Required
createdBy	string	Name of the user who created the comment.	Required
createdByUserID	string	Identity (guid) of the user who created the comment.	Required
updated	long	Comment update time in Unix time, seconds since January 1 1970.	Optional
updatedBy	string	Name of the user who updated the comment.	Optional
updatedByUserID	string	Identity (guid) of the user who updated the comment.	Optional

Measuring device alert

id	int	Identity of the measuring device that the alert refers to.	Required
alertID	int	Identity of the alert.	Required
description	string	Text that describes the alert.	Required
deviceAlertType	int	Type of alert: 0 = Device state 1 = Device measurement 2 = Reception 3 = Encryption	Required