ENERGINET
DataHub

Energinet DataHub A/S
Tonne Kjærsvej 65
DK-7000 Fredericia

+45 70 10 22 44
info@energinet.dk
VAT no. 39 31 50 41

Date:
March 21, 2022

Author:
SSG/JLI

CUSTOMER AND THIRD PARTY API FOR DATAHUB (ELOVERBLIK) – TECHNICAL DESCRIPTION

Document history

Version	Date	Description	Author(s)
1.0	01-05-2019	Initial version	Søren Søndergaard Janine Lindberg
2.0	22-11-2019	The following major changes apply: • Incorrect path for *Get authorizations* (section 2.7.2) has been corrected. • Information about main response structures have been added for *Get time series* (sections 1.7.8 and 2.7.6). • The mentioning of limits on the number of metering points have been removed from the description of all methods supporting multiple metering points in the request, since this has not been implemented. The same applies to the limit on result values for *Get time series* (sections 1.7.8 and 2.7.6). • In the response examples for *Get meter readings* in appendix B, meterNumber has been removed from the top-level structure and inserted into the Readings structure instead. • The JSON response example for *Get time series* in Appendix B has been corrected to specify that the Period structure can be repeated.	Janine Lindberg
3.0	11-12-2019	API location for pre-production environment has been changed, sections 1.2 and 2.2.	Janine Lindberg
4.0	27-12-2019	Spelling mistake in third party API URL for production environment has been corrected, section 2.2.	Janine Lindberg
5.0	09-01-2020	Incorrect paths for several methods have been corrected (sections 1.7.2, 1.7.3, 1.7.4, 1.7.5, 1.7.6, 1.7.7 and 2.7.3).	Janine Lindberg
6.0	10-11-2020	The term 'OAuth 2.0 tokens' has been replaced by 'bearer tokens' (sections 1.5 and 2.5).	Janine Lindberg
6.1	24-11-2021	Details of the functionality related to CPR match as part of the endpoints *Get metering points* and *Add relation based on CPR/CVR* have been specified (sections 1.7.2 and 1.7.3). Details of the functionality related to the *Get charges* endpoint have been specified (sections 1.7.7 and 2.7.5). Sections regarding swagger documentation and tool have been added (new sections 1.3 and 2.3). It has been specified that the *Get meter readings* endpoints will no longer be available by the end of 2021 (sections 1.7.9 and 2.7.7).	Janine Lindberg
6.2	21-03-2022	• Links to the Swagger documentation in sections 1.3 and 2.3 have been updated. • Appendix A has been deleted, and references to this appendix in sections 1.5 and 2.5 have been replaced with references to the Swagger documentation. • Appendix B has been deleted, and references to this appendix have been replaced with references to the Swagger documentation. • In sections 1.7.8 and 2.7.6 information regarding date and time format for period.timeInterval and period has been added in the response details table.	Janine Lindberg
6.3	05-10-2022	• Updated token expiry time (section 1.6 and 2.6) • Added new endpoint description (section 2.7.8)	Peter Gydesen

6.4	21-11-2022	• Added descriptions of new IsAlive endpoints • Updated endpoint paths • Updated various descriptions	Anders Blirup Worm
6.5	26-04-2024	• Removal of preproduction endpoints	Peter Gydesen
6.6	07-05-2024	• Added information about new IsAlive endpoint	Peter Gydesen

Table of contents

1. Customer API

1.1 Introduction

This API is intended to be used by customers (i.e. electricity consumers and producers) who want to access their own data from DataHub. To access the data the user must be authorized by use of a token (see section 1.6 for further information). When using the token all endpoints described in this chapter are accessible.

The following data can be requested:
1. List of metering points associated with the user (either linked or not linked), including selected metering point details (master data) (see section 1.7.2)
2. Extended list of details (master data) per metering point (see section 1.7.6)
3. Charge data per metering point (see section 1.7.7)
4. Time series per metering point (see section 1.7.8)
5. Meter readings per metering point (see section 1.7.9)

To request data mentioned in items 2-5 above, the metering points in question must first be actively linked to the user. There are two ways of creating links/relations to metering points:
1. Submit a list of metering points which are registered in DataHub to the user's CPR or CVR number retrievable from the supplied token (see section 1.7.3)
2. Submit a valid combination of metering point id and web access code (see section 1.7.4)

In addition to the token and data endpoints, there is an "IsAlive" endpoint. This does not require authentication and can be used to determine whether DataHub is operating normally, and the API is able to handle data requests.

1.2 API location

Environment	URL
Production environment	https://api.eloverblik.dk/customerapi/

1.3 Swagger documentation and tool

The relevant swagger documentation and tool can be found here:
https://api.eloverblik.dk/customerapi/index.html

1.4 Correlation id

It is possible to set a correlation id in the request header using the 'X-User-Correlation-ID' (with a UUID). When provided this id will follow the request and finally be returned via the response header. This id can be used for tracking the request. In parallel to this id there is another, internal id (also a UUID), which is returned in the response header as 'X-Correlation-ID'.

Please note that the *Get time series* response is subject to a different market message standard. Therefore, only one id (mRID) is returned in this response. This id is an internal id similar to the X-Correlation-ID.

1.5 Error codes and HTTP responses

A list of relevant error codes and HTTP status codes can be found in the Swagger documentation.

1.6   Tokens

Authentication and authorization is handled by using bearer tokens. To get started, a refresh token is required. A refresh token for customer API access can be created in the Eloverblik web portal after logging in as a private or business customer. The token is a long text string (JWT token), which must be copied and stored for use with the system that needs to access the API.

When a refresh token is obtained, the token endpoint can be accessed to create a short-lived data access token (valid for 24 hours). See section 1.7.1 for further information. For all data access the data access token needs to be supplied in the HTTP header:

Authorization: Bearer <data-access-token>

1.7   Endpoints

1.7.1 Get data access token

Path: /api/token
Parameters: None
HTTP verb: GET
Response: See example in the Swagger documentation.

Note: Include the refresh token in the HTTP header as mentioned in section 1.6.

1.7.2 Get metering points

This request is used for getting a list of metering points associated with a specific user (either private or business user). If the parameter includeAll is false (default), only metering points actively linked/related to the user (i.e. metering points with existing relations) are returned. If includeAll is true, the list of actively linked/related metering points will be merged with additional non-linked metering points registered in DataHub to the CPR or CVR number of the user. The CPR (private customers) or CVR (business customers) is retrieved by use of the supplied token. If the token is issued to a private customer, non-linked metering points are only returned if the CPR of the user, as well as the user's name as stated in the token, match data registered to metering points in DataHub.

Path: /api/meteringpoints/meteringpoints?includeAll={value}
Parameters:

Name	Type	Value
includeall	Boolean	true | false

HTTP verb: GET
Response: See example in Swagger documentation.

1.7.3 Add relation based on CPR/CVR

This request is used for linking one or more metering points to a user when the metering points are registered to the user's CPR or CVR in DataHub. The system will retrieve the CVR or CPR number related to the supplied token and verify if the supplied metering points are registered in DataHub to the retrieved CPR/CVR. If the token is issued to a private customer, not only the CPR is verified, but also the user's name as stated in the token must match the data registered to the supplied metering points in DataHub. If so, the relations are created. If not, the request is rejected.

The logical flow will be as follows:
1. First use the Get metering points endpoint (as described in section 1.7.2) to get a list of linked and non-linked metering points (i.e metering points with and without relations)
2. Then use the Add relation endpoint to link (i.e. create relations for) all or some of the non-linked metering points

Path: /api/meteringpoints/meteringpoint/relation/add
Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids
		See example in Swagger documentation

HTTP verb: POST
Response: String with the value of the metering point id. See example in Swagger documentation.

1.7.4 Add relation with web access code

This request is used for linking a metering point to a user by use of the web access code (WAC) for the metering point. The WAC is provided by the electricity supplier and can typically be found on the electricity bill. Adding relations by use of WAC is relevant for many metering points associated with private users/customers since many such metering points do not yet have CPR numbers registered in DataHub. Therefore, it is not possible to use the flow described in section 1.7.3. CPR numbers are typically added in DataHub when a customer moves or changes electricity supplier.

Path: /api/meteringpoints/meteringpoint/relation/add/{meteringPointId}/{webAccessCode}
Parameters:

Name	Type	Value
meteringPointId	string	Id of the metering point
webAccessCode	string	Web access code for metering point

HTTP verb: PUT
Response: String with the value of the metering point id. See example in the Swagger documentation.

1.7.5 Delete relation

This request is used for deleting an existing relation to a metering point.

Path: /api/meteringpoints/meteringpoint/relation/{meteringPointId}
Parameters:

Name	Type	Value
meteringPointId	string	Id of the metering point

HTTP verb: DELETE
Response: String with the value of the metering point id. See example in the Swagger documentation.

1.7.6 Get metering point details

This request is used for querying details (master data) for one or more (linked/related) metering points.

Path: /api/meteringpoints/meteringpoint/getdetails

Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation

HTTP verb: POST

Response: See example in the Swagger documentation.

1.7.7 Get charges

This request is used for querying charge data (subscriptions, tariffs and fees) for one or more (linked/related) metering points. Charges linked to the metering point at the time of the request or on any future date will be returned. However, past and future changes to an existing charge is not returned.

Path: /api/meteringpoints/meteringpoint/getcharges

Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation

HTTP verb: POST

Response: See example in the Swagger documentation.

1.7.8 Get time series

This request is used for querying time series for one or more (linked/related) metering points for a specified period and with a specified aggregation level.

Path: /api/meterdata/gettimeseries/{dateFrom}/{dateTo}/{aggregation}

Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation
dateFrom	string	YYYY-MM-DD
dateTo	string	YYYY-MM-DD
aggregation	string	Actual | Quarter | Hour | Day | Month | Year

HTTP verb: POST

Response: See example in the Swagger documentation.

Response details: The following table provides information about some of the main structures in the *Get time series* response.

Structure	Description
MyEnergyData_MarketDocument	One MyEnergyData_MarketDocument structure will be returned per metering point.
period.timeInterval	The structure specifies the total time interval for all time series in the specific MarketDocument structure. The time interval is expressed in UTC as specified in ISO 8601.
TimeSeries	If the metering point is non-profiled settled or flex settled (settlementMethod = E02 or D01), a maximum of 1 TimeSeries structure may be

	returned, containing all returned non-profiled energy quantities. If the metering point is profiled settled (settlementMethod = E01), a maximum of 2 TimeSeries structures may be returned – one structure containing non-profiled energy quantities and another containing profiled energy quantities.
MarketEvaluationPoint	This structure specifies the location where the time series are measured.
Period	For non-profiled energy quantities, the following applies: • 1 period per day will be returned if 15 minutes, hourly or daily resolution is returned • 1 period per month will be returned if monthly resolution is returned) • 1 period per year will be returned if yearly resolution is returned) For profiled energy quantities one period will be returned per energy quantity as registered in DataHub. Various period resolutions may apply. Periods are expressed in UTC as specified in ISO 8601.
Point	This structure contains 1-96 positions depending on the nature of the periods as described above.

1.7.9 Get meter readings

This request is used for querying meter readings for one or more (linked/related) metering points for a specified period.

Please note: Submission of meter readings to DataHub is no longer mandatory since end of 2021. Therefore, data may not be available for all metering points.

Path: /api/meterdata/getmeterreadings/{dateFrom}/{dateTo}
Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation
dateFrom	string	YYYY-MM-DD
dateTo	string	YYYY-MM-DD

HTTP verb: POST
Response: See example in the Swagger documentation.

1.7.10 IsAlive

Path: /api/isalive
Parameters: None
HTTP verb: GET
Response: True during normal operation. Will return HTTP 503 Service Unavailable if we are currently unable to handle requests.

Note: Does not require authentication. Responses are cached for 60 seconds at a time.

Path: /api/isalive
Parameters: None
HTTP verb: HEAD
Response: Header information on normal operation. Will return HTTP 503 Service Unavailable if we are currently unable to handle requests.

Note: Does not require authentication. Responses are cached for 60 seconds at a time.

2. Third party API

2.1 Introduction

This API enables third parties to access customers' data from DataHub based on authorizations (powers of attorney) granted by the customer. To gain access to the API the third party must:

- be registered as a third party to DataHub (further information can be found on Energinet's website: https://energinet.dk/El/Elmarkedet/Saadan-bliver-du-tredjepart).
- be authorized by use of a token (see section 2.6).

Authentication and authorization is handled by using tokens. Further information can be found in section 2.6. When using the token, all endpoints described in this chapter are accessible.

To retrieve actual data the third party must be authorized by customers to access data for specific metering points for specified periods of time. The third party must request access from the customer by a separate process which is not part of the API. [More information about this process will follow later. However, the process is expected to be similar to the existing process, for which documentation can be found on the Energinet website (see link above).]

The following data can be requested by use of the third party API:

1. List of authorizations (power of attorneys) (see section 2.7.2)
2. List of metering points, including selected metering point details (master data) (see section 2.7.3)
3. Extended list of details (master data) per metering point (see section 2.7.4)
4. Charge data per metering point (see section 2.7.5)
5. Time series per metering point (see section 2.7.6)
6. Meter readings per metering point (see section 2.7.7)

In addition to the token and data endpoints, there is an "IsAlive" endpoint. This does not require authentication and can be used to determine whether DataHub is operating normally and the API is able to handle data requests.

2.2 API location

Environment	URL
Production environment	https://api.eloverblik.dk/thirdpartyapi/

2.3 Swagger documentation and tool

The relevant swagger documentation and tool can be found here:
https://api.eloverblik.dk/thirdpartyapi/index.html

2.4 Correlation id

It is possible to set a correlation id in the request header using the 'X-User-Correlation-ID' (with a UUID). When provided, this id will follow the request and is finally returned via a response header. This id can be used for tracking the request. Parallel to this id there is another, internal id (Also a UUID), which is returned in the response header as 'X-Correlation-ID'.

Please note that the *Get time series* response is subject to a different market message standard. Therefore, only one id (mRID) is returned in this response. This id is an internal id similar to the X-Correlation-ID.

2.5   Error codes and HTTP responses

A list of relevant error codes and HTTP status codes can be found in the Swagger documentation.

2.6   Tokens

Authentication and authorization is handled by using bearer tokens. To get started, a refresh token is required. A refresh token for third party API access can be created in the Eloverblik portal after logging in as a registered third party. The token is a long text string (JWT token), which must be copied and stored for use with the system that needs to access the API.

When a refresh token is obtained, the token endpoint can be accessed to create a short-lived data access token (valid for 24 hours). See section 2.7.1 for further information. For all data access the data access token needs to be supplied in the HTTP header:

Authorization: Bearer <data-access-token>

2.7   Endpoints

2.7.1 Get data access token

Path: /api/token
Parameters: None
HTTP verb: GET
Response: See example in the Swagger documentation.

Note: Include the refresh token in the HTTP header as mentioned in section 2.6.

2.7.2 Get authorizations

This request is used for getting details about authorizations (powers of attorney) granted by customers. Only data regarding valid/active authorizations are returned. Data regarding deleted or expired authorizations is not returned.

Path: /api/authorization/authorizations
Parameters: None
HTTP verb: GET
Response: See example in the Swagger documentation.

2.7.3 Get metering points

This request is used for getting a list of metering points for which access has been granted. A filter value must be supplied to limit the result.
The following filter options are available:
- authorizationId: Returns a list of metering points included in a specific customer authorization
- customerCVR: Returns a list of metering points for which access has been granted by a specific customer (based on the CVR of the customer/granting party)
- customerKey: Returns a list of metering points which are linked to a specific customer key. The customer key can be supplied by the third party during the process for granting data access.

Path: /api/authorization/authorization/meteringpoints/{scope}/{identifier}

Parameters:

Name	Type	Value
scope	string	authorizationId | customerCVR | customerKey
identifier	string	Authorization Id, customer CVR or customer key

HTTP verb: GET

Response: See example in the Swagger documentation.

2.7.4 Get metering point details

This request is used for querying details (master data) for one or more metering points (for which access has been granted).

Path: /api/meteringpoint/getdetails

Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation

HTTP verb: POST

Response: See example in the Swagger documentation.

2.7.5 Get charges

This request is used for querying charge data (subscriptions and tariffs) for one or more metering points (for which access has been granted). Charges linked to the metering point at the time of the request or on any future date will be returned. However, past and future changes to an existing charge are not returned.

Path: /api/meteringpoint/getcharges

Parameters:

Name	Type	Value
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation

HTTP verb: POST

Response: See example in the Swagger documentation.

2.7.6 Get time series

This request is used for querying time series for one or more metering points (for which access has been granted) for a specified period and with a specified aggregation level.

Path: /api/meterdata/gettimeseries/{dateFrom}/{dateTo}/{aggregation}

Parameters:

Name	Type	Value
dateFrom	string	YYYY-MM-DD
dateTo	string	YYYY-MM-DD
aggregation	string	Actual | Quarter | Hour | Day | Month | Year
meteringPointIds	String array	List of metering point ids See example in the Swagger documentation

HTTP verb: POST

Response: See example in the Swagger documentation.

Response details: The following table provides information about some of the main structures in the *Get time series* response.

Structure	Description
MyEnergyData_MarketDocument	One MyEnergyData_MarketDocument structure will be returned per metering point.
period.timeInterval	The structure specifies the total time interval for all time series in the specific MarketDocument structure. The time interval is expressed in UTC as specified in ISO 8601.
TimeSeries	If the metering point is non-profiled settled or flex settled (settlementMethod = E02 or D01), a maximum of 1 TimeSeries structure may be returned, containing all returned non-profiled energy quantities. If the metering point is profiled settled (settlementMethod = E01), a maximum of 2 TimeSeries structures may be returned – one structure containing non-profiled energy quantities and another containing profiled energy quantities.
MarketEvaluationPoint	This structure specifies the location where the time series are measured.
Period	For non-profiled energy quantities, the following applies: • 1 period per day will be returned if 15 minutes, hourly or daily resolution is returned • 1 period per month will be returned if monthly resolution is returned) • 1 period per year will be returned if yearly resolution is returned) For profiled energy quantities one period will be returned per energy quantity as registered in DataHub. Various period resolutions may apply. Periods are expressed in UTC as specified in ISO 8601.
Point	This structure contains 1-96 positions depending on the nature of the periods as described above.

2.7.7 Get meter readings

This request is used for querying meter readings for one or more metering points (for which access has been granted) for a specified period.

Please note: Submission of meter readings to DataHub is no longer mandatory since end of 2021. Therefore, data may not be available for all metering points.

Path: /api/meterdata/getmeterreadings/{dateFrom}/{dateTo}
Parameters:

Name	Type	Value
dateFrom	string	YYYY-MM-DD
dateTo	string	YYYY-MM-DD
meteringPointIds	String array	List of metering point ids

		See example in the Swagger documentation

HTTP verb: POST

Response: See example in the Swagger documentation.

2.7.8 Get metering point ids

This request is used for getting a list of metering point ids for which access has been granted. A filter value must be supplied to limit the result.
The following filter options are available:

- authorizationId: Returns a list of metering point ids included in a specific customer authorization
- customerCVR: Returns a list of metering point ids for which access has been granted by a specific customer (based on the CVR of the customer/granting party)
- customerKey: Returns a list of metering point ids which are linked to a specific customer key. The customer key can be supplied by the third party during the process of granting access.

Path: /api/authorization/authorization/meteringpointids/{scope}/{identifier}

Parameters:

Name	Type	Value
scope	string	authorizationId | customerCVR | customerKey
identifier	string	Authorization Id, customer CVR or customer key

HTTP verb: GET

Response: See example in the Swagger documentation.