Data services user guide

Date: 08.01.2021 Version 1.0.3

Riga, 2021

CHANGE HISTORY

Modifier	Version	Date	Change description
Eduards Bezverhijs	1.0.0	18.09.2020	Initial document version created Services: • "get-object-list" • "get-object-consumption"
Eduards Bezverhijs	1.0.1	18.09.2020	Clarifications in the original version of the document
Tomass Liepnieks	1.0.2	26.11.2020	Clarifications in the original version of the document
Eduards Bezverhijs	1.0.3	08.01.2021	General clarifications and additions to the error list

TABLE OF CONTENT

INTRODUCTION

The latest generation of M2M (machine-to-machine) technologies enables automated data transfer without human intervention. This solution opens wide opportunities to improve the company's efficiency, as well as to develop new business opportunities.

Over time, additional web services will be added to the data services portfolio.

In order to use the services, the customer or their representative will need to have the ability to independently configure and consume the services offered in the chosen technology.

This document will describe the service specification and request examples.

The following Data Services (*SOA*) services are offered to customers:

Service name	Service description
get-object-list (receiving the list of available objects)	The customer or the customer's authorized representative obtains a list of available objects available to them.
get-object-consumption (obtaining the consumption of the object)	The customer or the customer's authorized representative obtains the consumption of the objects available to them.

DATA SERVICES CONCEPT

Initiation of data exchange

In order to start the data exchange, the customer must perform authorization on the https://www.e-st.lv portal and in the section "DATA SERVICES" select the objects involved in the data exchange, as well as get acquainted with the DATA SERVICE user guide.

Data exchange authorization / authentication

An API key (*APIKEY*) is used as the authentication / authorization mechanism. *APIKEY* essentially is a unique alphanumeric string to be used in service authorization and provides access to object data.

APIKEY can be created in the "DATA SERVICES" section of the https://www.e-st.lv portal.

An object can have only one *APIKEY* assigned to it, but if the customer wants, the same *APIKEY* can be used on several objects or groups of objects to group objects into lists, for example, according to the geographical location or some other principle convenient for the customer.

***APIKEY* is the only AUTHORIZATION mechanism for the DATA SERVICE. The Customer is fully responsible for its distribution to third parties who may use the *APIKEY* in data exchange on behalf of the Customer.**

Technically, *APIKEY* is used in conjunction with a so-called *Bearer* token, where the token value is *APIKEY*. See the examples attached to the specification for more details.

Data exchange format

Taking into account efficiency of data transmission, the data services are using the *REST / JSON* standard.

A simplified exchange format without using *JSON* schemas is used for data exchange, but the data element description specifies the format and whether the element is mandatory.

For the convenience and better understanding of the specification, customers and their representatives can consult request / response examples which are available as an attachments in *UTF-8*, *Unix EOL* format.

Service access endpoints

Access endpoint to the data service: https://services.e-st.lv

Note: *The URI of each API request is specified in particular API description section.*

It should be noted that successful data exchange always returns 200 series of *HTTP* codes, error cases are described in the section "Common error messages and error handling".

ADDITIONAL SOA SERVICE DESCRIPTION

get-object-list request

The purpose of this SOA service is to provide the customer or their representative with basic information about the objects which are available to them.

Service description

The service returns information about the object information available to the customer or their representative in accordance with the APIKEY assigned to the object.

Additional terms of use

There are no additional terms of use for this service.

Request URI

URI: `https://services.e-st.lv/m2m/get-object-list`

Request example

HTTP header: `Authorization: Bearer 123456789abcdefg123456789abcdefg123456789abcdefg123456789abcdefg`

URI: `https://services.e-st.lv/m2m/get-object-list`

Response example *(formatted)*

`get-object-list_response.json` - file (*Unix EOL*) is available as an attachment.

Request element details

Element name	Data type	Mandatory	Explanation
-	-	-	-

Response element details

Element name	Data type	Mandatory	Explanation
{}	*JSON type*	*Yes*	*Root level structure*
sDate	dateTime	Yes	Date when the data is available (value is in *XML dateTime* format). A start date of the service
cEIC	string	Yes	Customer EIC code. Customer identifier in ST systems
cName	string	Yes	Customer name/surname or company name
oList	*JSON array*	*Yes*	*List of objects*
{}	*JSON type*	*Yes*	*Composite element containing object information*
oEIC	string	Yes	Object EIC code. Object identifier in ST systems
oName	string	Yes	Object name
oStatus	string	Yes	Object connection status: • A – object is connected (active) • I – object is disconnected (not active)
oDate	dateTime	No	Date of last change of object counter status (value is in *XML dateTime* format)
oAddr	*JSON object*	*Yes*	*Composite element containing address information*
region	string	No	Address: region
country	string	Yes	Address: country
city	string	No	Address: city
county	string	No	Address: country
village	string	No	Address: village
street	string	No	Address: street
hNameNr	string	No	Address: house number / name
flatNr	string	No	Address: apartment (flat) number
pCode	string	No	Address: postal code
custAddrDet	string	No	Address: custom address detail. For use with non-standard addresses.
mpList	*JSON array*	*Yes*	*Metering point list*
{}	*JSON type*	*Yes*	*Composite element containing metering point information*
mpNr	string	Yes	Metering point number
mList	*JSON array*	*Yes*	*List of meters*
{}	*JSON type*	*Yes*	*Composite element containing meter information*

Element name	Data type	Mandatory	Explanation
mNr	string	Yes	Meter number

get-object-consumption request

The purpose of this SOA service is to provide the customer or their representative with the consumption information for the objects available to them.

Service description

The service returns data according to the APIKEY assigned to the object in addition to the rest of parameters for the data request.

The service provides the consumption information only for objects with smart meters. The service provides information about both the consumption used for billing and the actual consumption from the smart meter, which was successfully read the first time the meter was read.

Additional terms of use

Maximum available consumption data period: one year from the end of the current billing period.

If the request period exceeds one year, only one year of data, counting from the *dT* value element, is returned, i.e. for the last 365 days.

Request URI

URI: `https://services.e-st.lv/m2m/get-object-consumption?oEIC=___&mpNr=___&mNr=___&dF=___&dT=___`

Request example

HTTP header: `Authorization: Bearer 123456789abcdefg123456789abcdefg123456789abcdefg123456789abcdefg`

URI: `https://services.e-st.lv/m2m/get-object-consumption?oEIC=43Z-STO11111111K&mpNr=12345678&mNr=87654321&dF=2020-11-30T01:00:00&dT=2020-12-01T00:00:00`

Response example *(formatted)*

`get-object-consumption_response.json` - file (*Unix EOL*) is available as an attachment.

Request elements details

Element name	Data type	Mandatory	Explanation
oEIC	string	Yes	Object EIC code. Object identifier in ST systems
mpNr	string	No	Metering point number
mNr	string	No	Meter number
dF	dateTime	Yes	Start of the period for which the data is to be returned (value is in *XML dateTime* format) Note: if you want to get data for the full previous day, please specify the same *dF* and *dT* element values. Example: *dF* 2020-11-11T00:00:00 and *dT* 2020-11-11T00:00:00, the full 24h of data for 2020-11-11 will be returned.
dT	dateTime	Yes	End of period for which data is to be returned (value is in *XML dateTime* format)

Response elements details

Element name	Data type	Mandatory	Explanation
[]	*JSON array*	*Yes*	*Root level structure*
{}	*JSON type*	*Yes*	*Composite element containing metering point information*
mpNr	string	Yes	Metering point number
mList	*JSON array*	*Yes*	*List of meters*
{}	*JSON type*	*Yes*	*Composite element containing meter information*
mNr	string	Yes	Meter number
cList	*JSON array*	*Yes*	*Consumption list*
{}	*JSON type*	*Yes*	*Composite element containing consumption information*

Element name	Data type	Mandatory	Explanation
cDt	dateTime	Yes	Date and time of consumption **Important:** The date and time of consumption are provided in the response according to the general accepted principle of displaying consumption data in the conditions of the open electricity market. This means that the amount consumed in the current hour will be displayed as the value at the very beginning of the next hour. For example, the consumption listed from 00-01 is returned as *"cDt": "2020-11-11T01:00:00.000+02:00"*, while the consumption listed from 01-02 is displayed as *"cDt": "2020-11-11T02:00:00.000+02:00"*, etc. The last hour or consumption listed from 23-24 is returned at the first hour of the next day *"cDt": "2020-11-12T00:00:00.000+02:00".*
cVR	decimal	Yes	First time read consumptions from meter
cVRSt	string	No	Consumption status: • C – data is not fully available or incorrect • D – consumption adjusted according to the difference of meter readings, there have been deficiencies in the load profile readings, the meter has not been read, consumption at the object is written off according to the average consumption (the amount of consumption does not match the total consumption of the period) • M – consumption was adjusted (consumption corrected in result of rounding error) • E – consumption was corrected by meter reading IS • N – communication error while connecting to the meter • U – value is not usable • CD or other combination of previously mentioned statuses, multiple reasons together
cVV	decimal	No	Consumption value which is used for billing

SHARED ELEMENTS STRUCTURES

The structure of the service error handling element

In case of errors, the services will return the response in the following structure. It should be noted that all requests that result in an error will return a series of 400/500 *HTTP* codes.

Details of error structure

Element name	Data type	Mandatory	Explanation
{}	*JSON type*	*Yes*	*First level structure (root)*
title	string	Yes	Error description element
invalid-params	*JSON array*	*Yes*	*Error parameter array*
{}	*JSON type*	*Yes*	*Composite element containing error information*
name	string	Yes	Error code / invalid element name
reason	string	Yes	Error text / description

COMMON ERROR MESSAGES AND ERROR HANDLING

In customer services, error processing takes place with the help of a separate error structure element.

Each service has its own error messages. If error handling is specific to a particular service, explanations of error codes and messages are provided with the service description.

In other cases, read the error message and act accordingly.

Common error messages

HTTP code		Error code	Error description
400	Bad request	M2M-ERR-004	Unexpected error while processing message.
400	Bad request	M2M-ERR-007	Input parameter "$VALUE" is not specified
400	Bad request	M2M-ERR-009	Unable to find customer with EIC "$VALUE".
400	Bad request	M2M-ERR-010	Unable to find object with EIC "$VALUE".
400	Bad request	M2M-ERR-011	The $VALUE date is not in the correct format.
400	Bad request	M2M-ERR-012	Start date must be in the past.
400	Bad request	M2M-ERR-013	End date must be less then start date.
400	Bad request	M2M-ERR-014	Parameter "$VALUE" is not in the correct format.
401	Error validating service security parameters	M2M-ERR-003	APIKey format is not valid
401	Error validating service security parameters	M2M-ERR-005	Parameter $VALUE is mandatory.
401	Error validating service security parameters	M2M-ERR-006	APIKey is not valid.
500	Internal server error	M2M-ERR-004	Unexpected error while processing security.
500	Internal server error	M2M-ERR-004	Unexpected error while processing message.

In case if you have any questions or concerns, please contact us: sistemas@sadalestikls.lv