DH API documentation for public supplier

Content

1 Documentation version history 3
2 Preface 7
3 Definitions and abbreviations 8
4 Environments 9
5 Digital certificates 11
6 Recommendation for API client 12
6.1 ASYNC 12
6.1.1 DataHub order processing retry policy 16
6.1.2 Order status flows 16
6.1.3 Recommendations 17
6.2 SYNC 17
6.3 JSON request logic 20
6.4 "Net billing" process 21
6.4.1 Getting "Net billing" prosumer graph 21
6.4.2 Correcting "Net billing" graph 25
7 DataHub Gatewy API documentation 31
7.1 Order controller 32
7.1.1 POST /gateway/public-supplier/order/list 32
7.1.2 GET /gateway/public-supplier/order/{orderId}/count 37
7.1.3 POST /gateway/public-supplier/order/data-hr-15min-obj-lvl 40
7.1.4 GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl 46
7.1.5 POST /gateway/public-supplier/order/data-hr-15min-history-changes 50
7.1.6 GET /gateway/public-supplier/order/{orderId}/data-hr-15min-history-changes 55
7.1.7 POST /gateway/public-supplier/order/balance-data 58
7.1.8 GET /gateway/public-supplier/order/{orderId}/balance-data 61

7.1.9 POST /gateway/public-supplier/order/balance-by-generation-type 65
7.1.10 GET /gateway/public-supplier/order/{orderId}/balance-by-generation-type 68
7.1.11 POST /gateway/public-supplier/order/balance-data-by-contract-type 72
7.1.12 GET /gateway/public-supplier/order/{orderId}/balance-data-by-contract-type 74
7.2 Object controller 77
7.2.1 POST /gateway/public-supplier/object/all/active/list 77

1 Documentation version history

The table below provides information on document version history:

Version	Date	Description
1.0.0	2023-12-06	Initial document version.
1.0.1	2024-02-14	Added new methods: • POST/gateway/public-supplier/order/data-hr-15min-history-changes • GET/gateway/public-supplier/order/{orderId}/data-hr-15min-history-changes In method POST/gateway/public-supplier/order/data-hr-15min-obj-lvl: • Added new attributes intervalData, intervalDataRecalculation, intervalDataDetailed in JSON request • Removed rule with error code 1005 • Added new rules with error codes 2026, 2027, 2028 In method GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl: • Added new attributes graphVersion, powerPlantObjectNumber, powerPlantType, usageType in JSON response
1.0.2	2024-03-01	Added new methods: • POST/gateway/public-supplier/order/balance-data • GET/gateway/public-supplier/order/{orderId}/balance-data In the method POST /gateway/public-supplier/order/list • added new orderType "balance-data"
1.0.3	2024-03-04	Added new methods: • POST/gateway/public-supplier/order/balance-by-generation-type • GET/gateway/public-supplier/order/{orderId}/balance-by-generation-type In the method POST /gateway/public-supplier/order/list • added new orderType "balance-by-generation-type"

1.0.4	2024-04-03	Added rule with 2015 error code in method: • POST /gateway/public-supplier/order/data-hr-15min-history-changes
1.0.5	2024-04-08	Added rule with 2030 error code in method: • POST /gateway/public-supplier/order/data-hr-15min-obj-lvl
1.0.6	2024-04-08	In method POST /gateway/public-supplier/order/ balance-by-generation-type: • updated request attribute generationType to list • updated obligation value to “not required“
1.0.7	2024-04-08	Added examples of generation changes (response attribute reasons) in method: • GET /gateway/public-supplier/order/{orderId}/data-hr-15min-history-changes
1.0.8	2024-04-25	In method POST /gateway/public-supplier/order/data-hr-15min-history-changes: • in JSON request deleted attribute dateTo • deleted rule with errror code 1002
1.0.9	2024-04-26	Updated rule message with error code 1008 in methods: • POST /gateway/public-supplier/order/data-hr-15min-obj-lvl • POST /gateway/public-supplier/order/data-hr-15min-history-changes • POST /gateway/public-supplier/order/balance-data • POST /gateway/public-supplier/order/balance-by-generation-type
1.0.10	2024-04-29	In method POST /gateway/public-supplier/order/data-hr-15min-history-changes: • removed rule with error code 2015 • added new rule with error code 2031
1.0.11	2024-05-15	Updated rule description with error code 2030 in method: • POST/gateway/public-supplier/order/v2/data-hr-15min-obj-lvl
1.0.12	2024-05-20	Added rule with error code 2032 in method: • POST /gateway/public-supplier/order/data-hr-15min-obj-lvl

1.0.13	2024-05-29	Added rule with 2015 error code in methods: • POST /gateway/public-supplier/order/balance-data • POST /gateway/public-supplier/order/balance-by-generation-type
1.0.14	2024-06-05	Changed location of graphVersion attribute in method: • GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl
1.0.15	2024-06-05	The data type value of the attribute amount has been updated in method: • GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl
1.0.16	2024-06-12	Added recommendations for the "Net billing" process and SYNC.
1.0.17	2024-07-01	Removed error with error code 1003 in method (only in documentation): • POST /gateway/public-supplier/order/data-hr-15min-history-changes
1.0.18	2024-07-01	Removed errors with error code 2012, 2013 and added 2033 in method: • POST /gateway/public-supplier/order/data-hr-15min-history-changes
1.0.19	2024-09-04	Added new values of generationType and the possible combinations of the generation Category attribute have been removed in methods: • POST /gateway/public-supplier/order/ balance-by-generation-type • GET /gateway/public-supplier/order/{orderId}/balance-by-generation-type
1.0.20	2024-09-12	Added new values of powerPlantType in methods: • GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl
1.0.21	2025-03-03	Added new methods: • POST /gateway/public-supplier/order/balance-data-by-contract-type • GET /gateway/public-supplier/order/{orderId}/balance-data-by-contract-type In method POST /gateway/public-supplier/order/list added new orderType meaning.

1.0.22	2025-07-03	A new method POST /gateway/public-supplier/object/all/active/list was added to the new Object Controller.

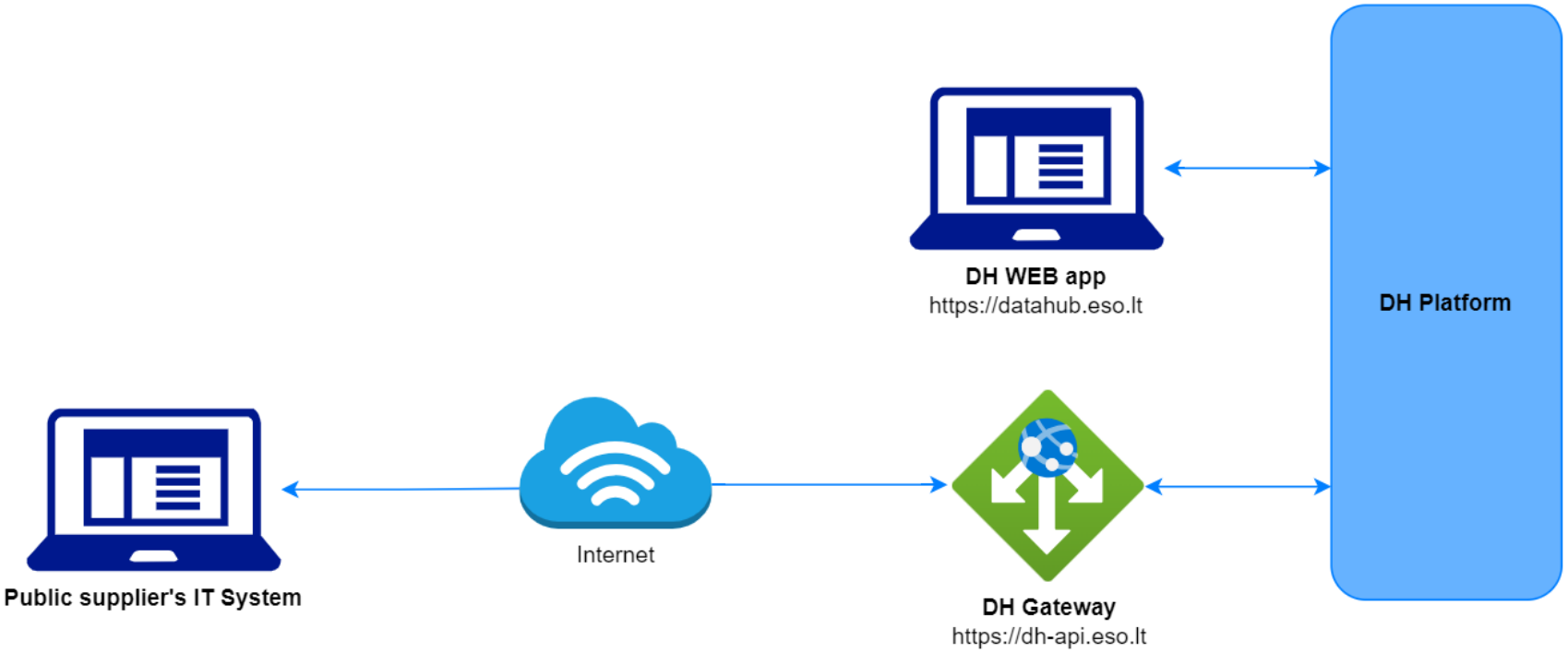
Note: Changes in table marked in white are already deployed, marked in green will be deployed in near future.

2 Preface

The Common Data Exchange Platform (hereinafter referred as DH Platform) Gateway is a component enabling public suppliers to directly access DH Platform from within their IT systems and thus helps perform their activities more efficiently.

DH Gateway provides open standards-based interfaces allowing public suppliers themselves (or with outside assistance) integrate their IT systems with DH Platform.

This document provides technical information on DH Gateway interfaces which is needed to integrate public suppliers' information systems with DH Platform.



3 Definitions and abbreviations

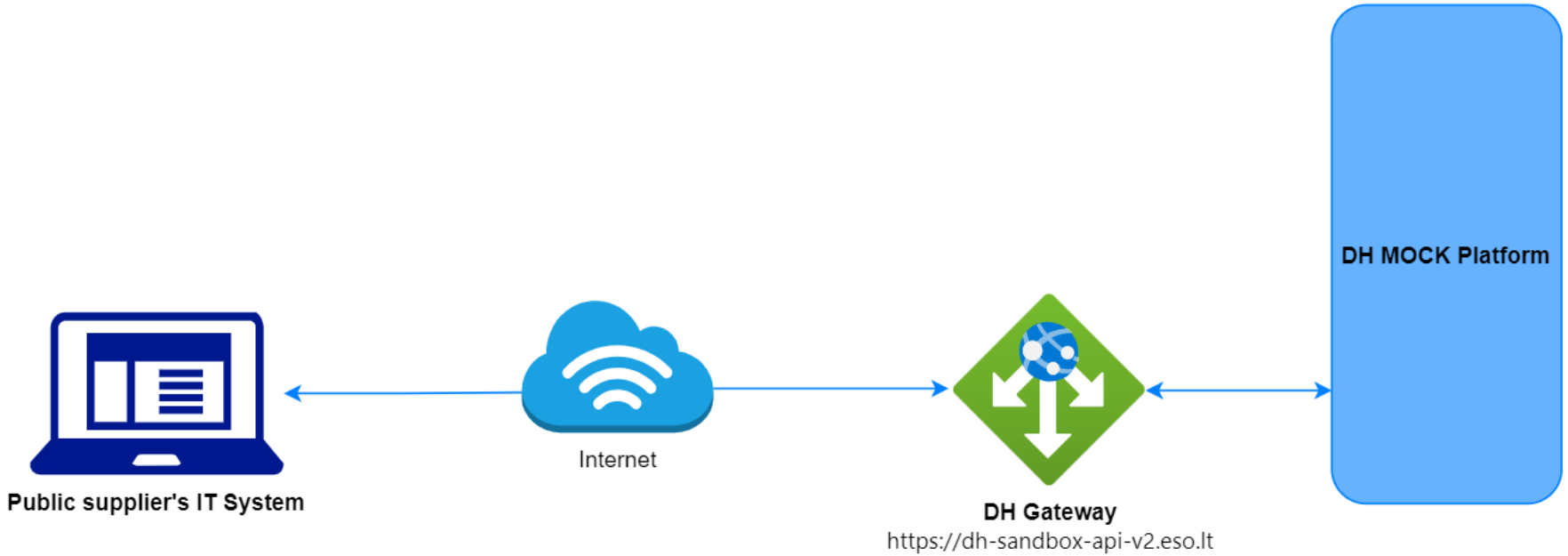
Definition / abbreviation	Description
DH, DH Platform, DataHub	The Common Data Exchange Platform.
DH Gateway	DH Platform component enabling public supplier’s IT systems to directly access the platform and achieve a higher-level degree of process automation.
API	Application Programming Interface
DSO, ESO	Energy distribution system operator – AB „Energijos skirstymo operatorius“.
Object	A site where electricity consumption takes place.
DH WEB app	It is a web application that provides a graphical user interface (GUI) for working with the DataHub system.
Value "N"	No
Value "Y"	Yes

4 Environments

There are two DH Gateway environments the public supplier might access:

- “Sandbox” environment
- Production environment

DH Sandbox environment made of Mock API Gateway with mock requests and responses (scenarios). There is no connection to database or any data source, all possible requests and answers are hard coded into mock API source code and has no data selection logic or rules. This data is real depersonalized data from DSO customers. Sandbox requests and responses scenarios will be provided in additional document, and it should be used just for preparation to integrate with DH production API environment or testing purposes.



DH Platform also has WEB interface, which is connected to DH Production Gateway. All environments are provided in the table:

Environment	Swagger Link	WEB Interface
Production	https://dh-api.eso.lt/swagger-ui.html	https://datahub.eso.lt/
Sandbox	https://dh-sandbox-api-v2.eso.lt/swagger-ui.html#/	-

5 Digital certificates

In both the testing and production environments of the DH Gateway component, the identity of the public supplier is established using a TOKEN, which the public supplier’s information system must provide each time the DH Gateway network service is called.

To get started:

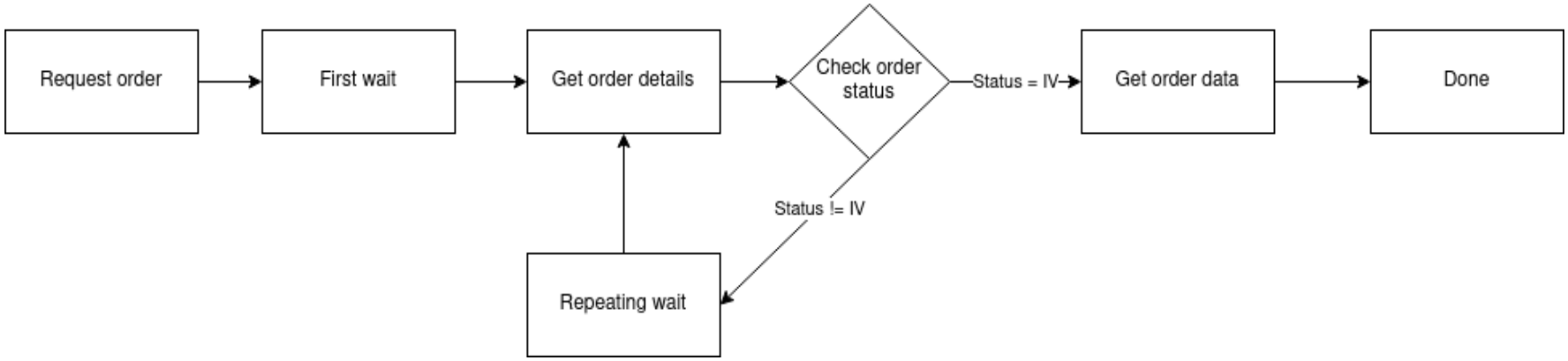
1. The DSO responsible person sends the JWT key (JSON Web Token) to be used with each request to the DH API.
2. To make requests to the DH Gateway API - the TOKEN submission in the case of curl takes place.

6 Recommendation for API client

6.1 ASYNC

Async pattern is mainly used for data orders: https://dh-api.eso.lt/swagger-ui/index.html?urls.primaryName=public-supplier#/public-supplier-order-controller

Client side should implement following process with steps:



Step descriptions

Step name	Description	Endpoint	Request example	Response example

Request order	Submit new data order. Request will return order id which will be used in other steps for getting order details and order data.	POST /gateway/public-supplier/order/**yyyyyyyyyy**
		where **yyyyyyyyyy** is order type:
		• data-hr-15min-obj-lvl
			POST /gateway/public-supplier/order/ data-hr-15min-obj-lvl
			Body:
			{
			"consumptionCategories" : [
			"P+"
			],
			"dateFrom": "2023-11-01",
				HTTP status 201
				{
				"orderId": 10000001
				}

			"dateTo": "2023-11-30", "interval": "HOUR", "objectNumbers": ["11111111", "22222222"] }	
First wait	Wait for some period of time after order submission. This step is needed because after order request it takes some time to process it and there is no reason to try get status immediately after order submission. First wait duration depends on order type and parameters. If order collects more data, it can take minutes to prepare data. For duration recommendations look at Recommendations.			

Get order details	Request to get order details. This request is needed to get order latest status which is stored in field "latestStatus".
	Posible values for "latestStatus":
	• P - Submitted order
	• V - Order in progress
	• IV - Order is finished and data are prepared.
	• K - Order has errors
		POST/gateway/public-supplier/order/list	POST /gateway/public-supplier/order/list
			{
			"orderId": 10000001
			}
				HTTP status 200
				[
				{
				"orderId": 10000001,
				"orderType": "data-hr-15min-obj-lvl",
				"submittedDate": "2023-12-07T08:49:29.117",
				"dateFrom": "2023-11-01",
				"dateTo": "2023-11-30",
				"orderParameters": "{\"consumptionCategories\“:[\"P+\"],\"objectNumbers\":[\"111111

				11\",\"22222222\"],\"interval\":\"HOUR\"}", "latestStatus": "V", "statusDate": "2023-12-07T08:49:30.446", "expireDate": "null, "auto": false, "userName": "PUBLIC" }]
Check order status	Logic operation to check order "latestStatus" field value. If value equals to "IV" it means that order data is prepared. Otherwise order data is not ready algorithm should go to step "Repeating wait".			
Repeating wait	Wait for some period of time after order status check when status was not equal to "IV". This step is needed because repetitive status check without wait can do unneeded load to DH system. For duration recommendations look at Recommendations.			

Get order data	Get order data.
	
	Note: If order has too many data, then pagination should be used. Default and max page size is 10 000 records (usually it's objects).
		GET/gateway/public-supplier/order/**zzzzzzzz**/**yyyyyyyyyy?**first=**oooooo**&count=**ssssss**
		
		where:
		
		• **zzzzzzzz** is order Id
		• **oooooo** is offset position
		• **ssssss** is page size
		
		yyyyyyyyyy is order type:
		
		• data-hr-15min-obj-lvl
			GET/gateway/public-supplier/order/10000001/ data-hr-15min-obj-lvl?first=0&count=10000	HTTP status 200 with order data in JSON format.
				
				If order content is empty get method will return HTTP status 400 with message.
				
				{
				"code": 2018,
				"text": "There is no data for the selected search parameters, the response is empty."
				}

6.1.1 DataHub order processing retry policy

If any issues appear during order data processing stage the process stops, and order gets status K. DH uses retry policy for all orders with status K.

- Retries order process after 5 minutes.
- Retries order process 300 times.
- For failed orders retry policy will be working in total 25 hours (5 min * 300).
- Retry policy will stop work after 25 hours and order will be left with status K.

This is needed because issues can appear in data preparation stage of couple reasons:

- DH technical problem - for example one of DH integrations was down or contract was changed, data integrity violations and etc.
- Incompatible business logic - for example order got into not defined use case and use case should be adopted to order.

In most cases order processing retry will solve problem. But there are cases like "Incompatible business logic" when additional human interaction is needed to finish order job. We are tracking such an order and fixing them, but fixing might take some hours or even days. So, some orders might not be completed and left in status K.

6.1.2 Order status flows

There are three possible order status flows:

Flow	Description
P → V → IV	This is normal status flow.
P → V → K → IV	This is flow when issues appear during data preparation, but later problem was fixed.
P → V → K	This is flow when issues appear during data preparation and problem was not fixed during DH retry policy time.

Order execution duration depends on multiply factors:

- Order type - different order types use different integration services some of them are faster some of them are slower.
- Order parameters - order parameters describe how much data will be generated. Bigger order periods and bigger object quantity will be generated longer.
- Order quantity in queue. If public supplier creates too many orders, they will be generated parallelly and will take more time to finish them all.
- Failures - Errors during order data preparation will trigger retry policy so order generation will take more time as usually. Sometimes it will be not generated at all.

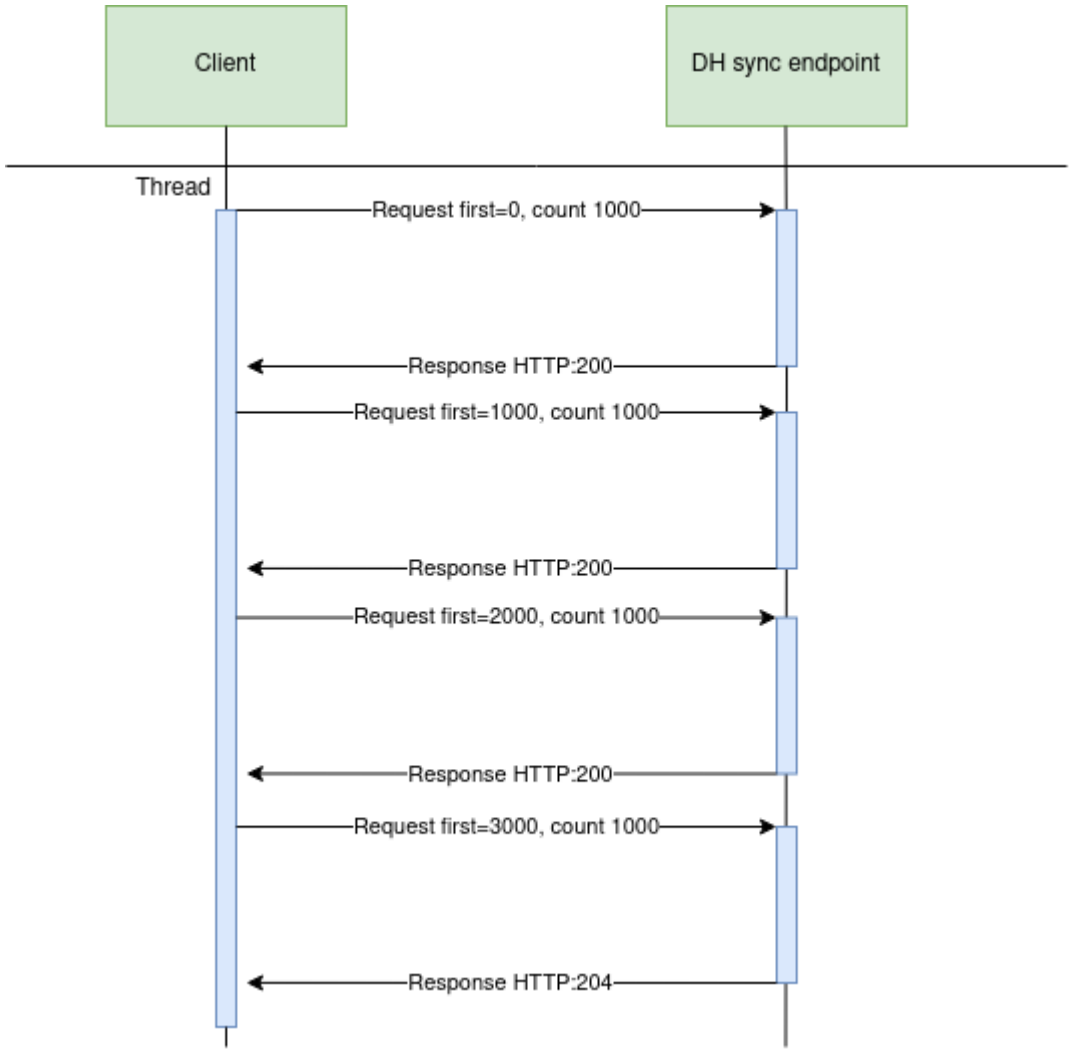
6.1.3 Recommendations

1. For better performance "Request order" can be implemented as separate process which is able to create multiple orders.
2. For better performance "Get order details" can be implemented as separate process which is able to get details of multiple orders.
3. For better performance "Get order data" can be implemented as separate process which is able to get order data of multiple orders.
4. For better performance process parallelization could be used but with max 3 threads.
5. Any HTTP request which returns 5xx status can be retried.
6. Any HTTP request which returns 4xx status should stop process because where are business error and manual handling should be used. Except for the step "Get order data" and error "code": 2018, "text": "There is no data for the selected search parameters, the response is empty." It means that order data preparation is finished, and order is empty.
7. Step "Request order" and other steps should have separate retries. Get order data on failure should not trigger Request order one more time.
8. It's up to client to decide how long the "First wait" duration can be but it shouldn't be less than 1 second.
9. It's up to client to decide how long the "Repeating wait" duration can be but it shouldn't be less than 1 second.
10. Use fixed number of times for status check. After 25 hours DH order retry policy will stop working and order will be left in status K. So, it reasonable to have number of times equal ((25 hours) / ("Repeating wait" duration in hours)).
11. Do not recreate orders when orders got status K. DH retry policy will try to generate it later or DH team member interaction is needed to finish order. Client-side solutions will not help to solve status K.

6.2 SYNC

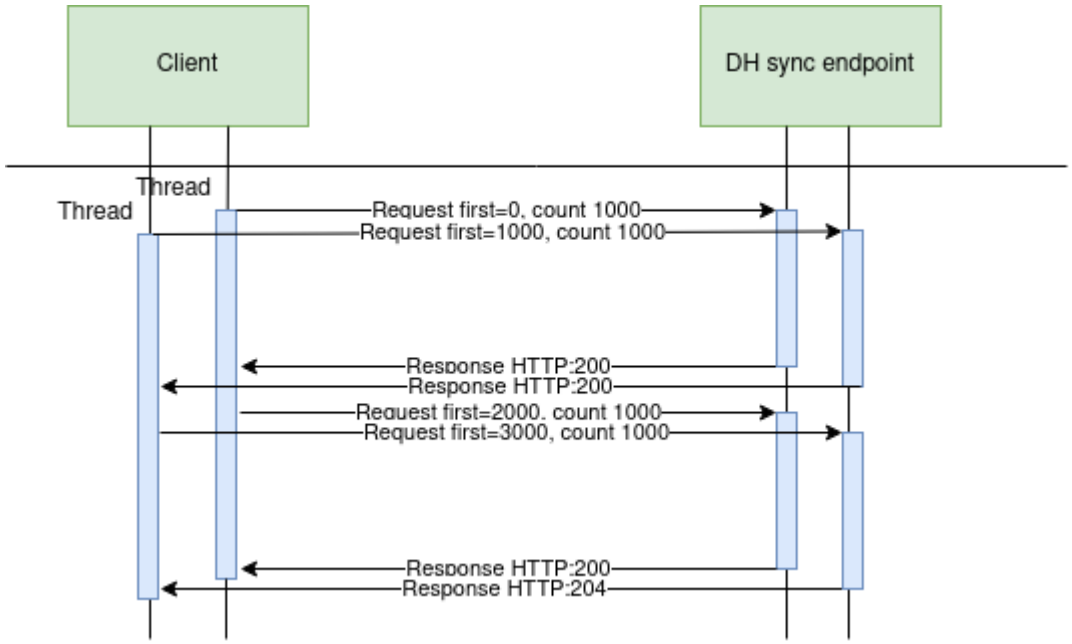
Sequential approach with pagination:

Should be default option



Parallel approach with pagination

Could be used when import time is not reasonable. But parallel requests should be limited.



Recommendations for DH client:

1. Parameter for page size control
2. Parameter for parallel thread amount control
3. Sequentially process should be default, but if import takes too much time, parallel process could be used but with max 3 threads.
4. Page size can be calculated by formula PS = PST/THRA, where PS - page size, PST - page size with which request takes less than 15 seconds to execute, THRA - thread amount. But not bigger than 10 000 records.
5. Retry on HTTP statuses: 429, 5xx
6. Retry should restart failed request only (not whole import process)
7. Retry interval should be not less than 5 seconds.
8. Import process should be able to continue process after failure

Motivation to have these features

1. Performance issues can arise accidentally and to solve them can take some time. So, during performance problem page size and thread amount control could help.
2. DH has planned and unplanned deployments also can be incidents and during them datahub services might be unavailable. In these cases, from client side is nice to have fallback process which retries failed requests and continues process after service become available again.
3. In the future throttling by independent supplier will be introduced and some of requests could end up with http status 429 (too many requests) so retry process also will be helpful hear.

6.3 JSON request logic

JSON field usage in requests by type:

Type	Example	Is value provided	Request result
integer	orderId: null	No	All orders.
integer	orderId: 4587125	Yes	Order with ID 4587125.
dateTime	submittedDateFrom: null	No	All orders.
dateTime	submittedDateFrom: ""	Yes	Framework validation error because provided value is not matching date format.
dateTime	submittedDateFrom: "2023-01-01"	Yes	All orders with were submitted date greater than 2023-01-01.
list	latestStatuses: null	No	All orders with all statuses.
list	latestStatuses: []	Yes	Empty list because provided latestStatuses list is not matching any latest status.
list	latestStatuses: [""] or latestStatuses: ["", ""]	Yes	Framework validation error because provided value is not matching statuses.
list	latestStatuses: [null] or latestStatuses: [null, null]	Yes	Empty list because provided latestStatuses list is not matching any latest status.
list	latestStatuses: ["IV"] or latestStatuses: ["IV", "V"]	Yes	Orders with statuses IV or V.
boolean	auto: null	No	All orders because criteria were not given.
boolean	auto: ""	No	Validation error because invalid Boolean value was given.

Type	Example	Is value provided	Request result
boolean	auto: "NOT BOOLEAN"	No	Validation error because invalid Boolean value was given.
boolean	auto: "false"	Yes	Orders which were ordered not automatically.

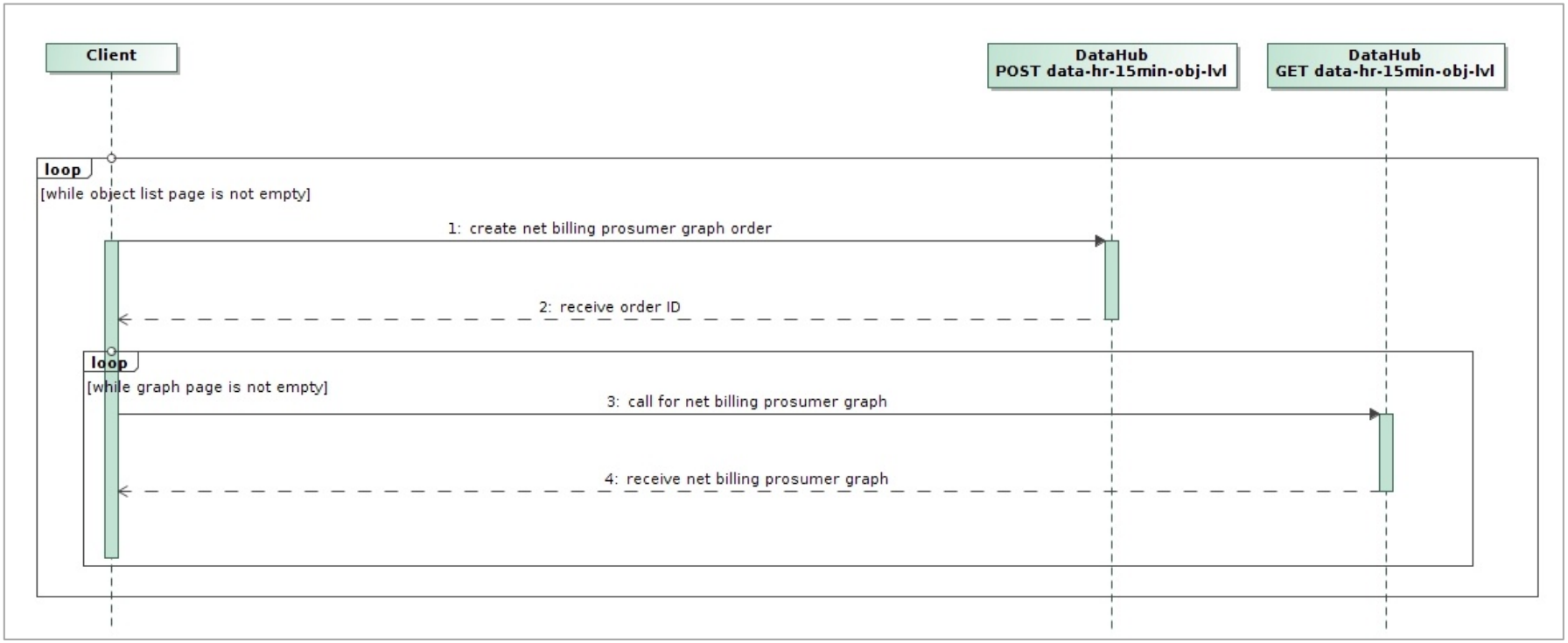
If field value is not provided, then field criteria shouldn’t be added to query and all lists should be returned.

6.4 "Net billing" process

"Net billing" process consist of 2 parts:

1. Getting "Net billing" prosumer graph
2. Correcting "Net billing" graph

6.4.1 Getting "Net billing" prosumer graph



Step	Name	Description	End-point	Request / Response example

1	create "Net billing" prosumer graph order	Create order which will prepare a prosumer graph.	POST /gateway/public-supplier/order/data-hr-15min-obj-lvl	**Request body**
				
				{
				"consumptionCategories": [
				"P+", "P-"
				],
				"dateFrom": "2024-05-01",
				"dateTo": "2024-05-31",
				"interval": "HOUR",
				"netBilling": {
				"intervalData": true,

				"intervalDataDetailed": true, "intervalDataRecalculation": false }, "objectNumbers": ["4565657", ...] } Where: • consumptionCategory - possible values: o P+ for consumption graph. o P- for generation graph. • dateFrom - the first day of billing period. • dateTo - the last day for billing period. • interval - possible values: o HOUR for hour graph. o QUARTER for 15-minute graph. • intervalData - should be true for "Net billing" process. • intervalDataDetailed - possible values: o true - if needed generation graph detailed with power plants. o false - if needed common generation graph without details. • intervalDataRecalculation - in this step should be false because we need to get graph as is in current time. • objectNumbers - list of object numbers which has "NET_BILLING" accounting scheme.
2	receive order ID	Step 1 will return order ID.		**Response body** {

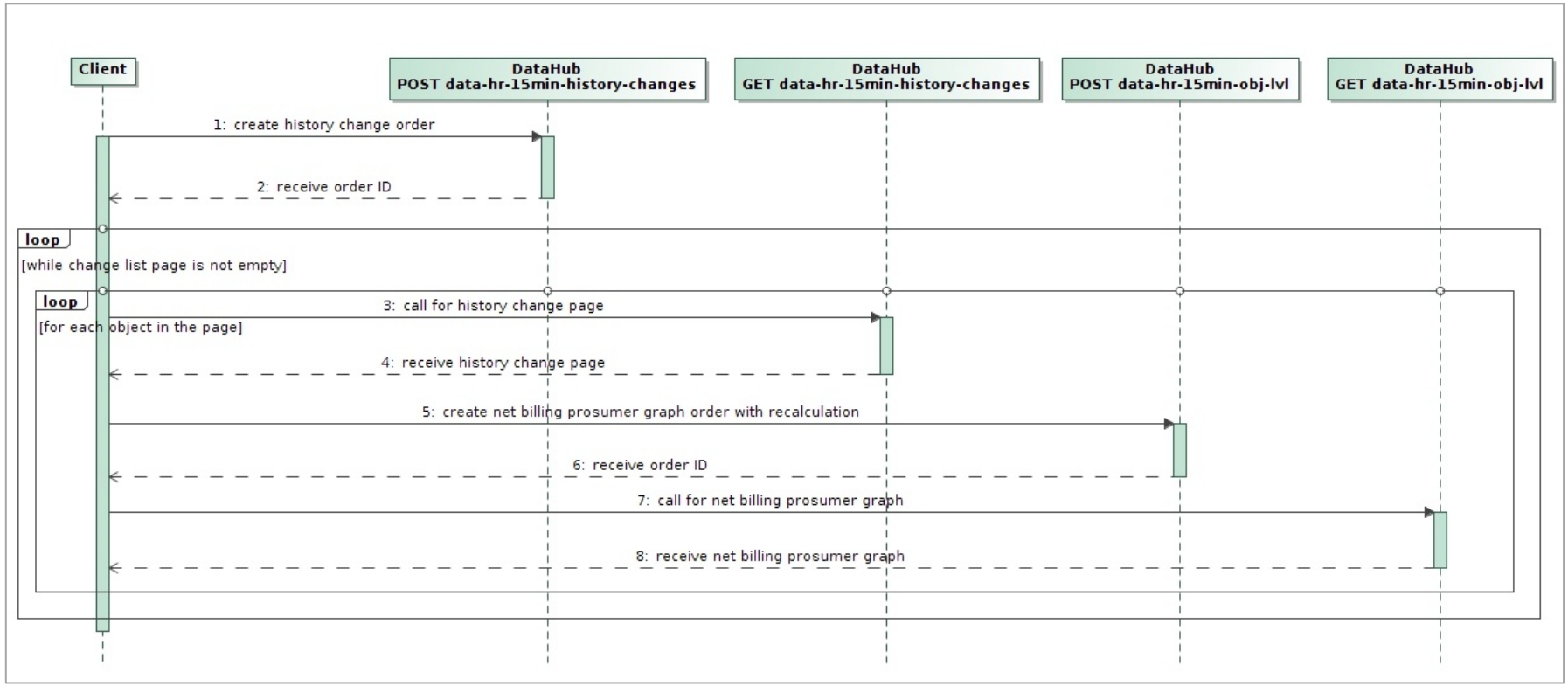
				"orderId": 1000001 }
3	call for "Net billing" prosumer graph	After successful "Net billing" prosumer graph generation. Graph should be read page by page. **Note**: • for more details on how to work with orders look in Async. • for more details on how to use pagination look in Sync.	GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl Where: • {orderId} - Order ID from step 2. • header: first = 0 • header: count: 10 Recommended page size 10 because graph will be generated for the whole month and one object will contain a lot of data.	

4	receive "Net billing" prosumer graph	Receive prosumer graphs.
		Note: usage type can be:
		• **B - Billing**. It means that final data version is captured from DH side and all next changes with data will be notified via history change report **data-hr-15min-history-changes**. This is recommended data status for "Net billing". Billing data will be captured and available:
		  ○ From 09:00 AM on 2th working day of each month.
		• **D - Daily**. it means that final version of data is not captured yet from DH side and can be changed later without any notification. This type of graph also can be used for "Net billing" but with additional process from client side. Client should periodically repeat steps from 1 to 4 for objects which had usage type D until will get usage type B. This
				Response body
				
				[
				{
				"consumptionCategories": [
				{
				"consumptionCategory": "P-",
				"consumptions": [
				{
				"amount": 45,
				"consumptionTime": "2024-05-10T18:00:000",
				"graphVersion": "2024-05-10T18:00:00.000",
				"usageType": "B",
				"valueType": "VAL"
				}
				],
				"powerPlantObjectNumber": "45654654",

		process might be needed in cases when Billing data for some reasons could not be prepared on time from DH side.		"powerPlantType": "A" }, ...], "objectBsId": 0, "objectNumber": "4565657", ... }]

6.4.2 Correcting "Net billing" graph

This process is about how identify objects which might have changes in previous billing periods and how to renew graphs.



Step	Name	Description	End-point	Request / Response example

1	create history change order	Create order which will prepare a list of "Net billing" objects which had data change events during current billing period and these changes did impact in previous accounting periods.	POST /gateway/public-supplier/order/data-hr-15min-history-changes	**Request body**
				
				{
				"dateFrom": "2024-05-01"
				}
				
				Where:
				• dateFrom - the first day of billing period
				
				Note: report will find all events which did impact on previous periods from dateFrom to current time.

2	receive order ID	Step 1 will return order ID.		**Response body**
				
				{
				"orderId": 1000002
				}

3	call for history change page	After successful history change order generation. Object list should be read page by page. **Note**: • for more details on how to work with orders look in Async. • for more details on how to use pagination look in Sync.	GET /gateway/public-supplier/order/{orderId}/data-hr-15min-history-changes Where: • {orderId} - Order ID from step 2. • header: first = 0 • header: count: 500	

4	receive history change page	Receive object list which has changes in previous billing periods and graph information could be recalculated and renewed. If result is empty it means that no changes was made in previous periods and actual data is correct.
		
		Note:
		
		History change order will tell which object have changes but it will not give guarantee that this change have impact in the "Net billing" balance. Impact will be known after step 8 and after compare between old graph version and recalculated version will be done.
			Recommended page size 500 because same size is the max size of object list in the next steps.	**Response body**
				
				[
				{
				"objectNumber": "4565657",
				"periodsWithChanges": [
				{
				"billingPeriod": "2024-01",
				"reasons": [
				"GENERATION_CHANGE"
				]
				}
				],
				....
				}
				]
				Where:
				• objectNumber - number of the object which has impact in history data.
				• periodsWithChanges - a list of object periods which should be recalculated and renewed because they have changes.
				• reasons - posible values:
				○ GENERATION_CHANGE - means that procumer graph have updates.
				○ OWNER_CHANGE - means retroactive owner change.
				○ SUPPLIER_CHANGE - means retroactive supplier change.
				○ SCHEMA_CHANGE - means retroactive accounting schema change.

5	create "Net billing" prosumer graph order with recalculation	Create order which recalculate and prepare new prosumer graph.
		
		Note:
		• order with recalculation should be created separately for each object from step 4.
			POST /gateway/public-supplier/order/data-hr-15min-obj-lvl	**Request body**
				
				{
				"consumptionCategories": [
				"P+", "P-"
				],
				"dateFrom": "2024-05-01",
				"dateTo": "2024-05-31",
				"interval": "HOUR",
				"netBilling": {
				"intervalData": true,
				"intervalDataDetailed": true,
				"intervalDataRecalculation": true
				},
				"objectNumbers": [
				"4565657"
				]
				}
				Where:
				• consumptionCategory - possible values:
				○ P+ for consumption graph.
				○ P- for generation graph.
				• dateFrom - the first day of billing period.
				• dateTo - the last day for billing period.
				• interval - possible values:
				○ HOUR for hour graph.
				○ QUARTER for 15-minute graph.
				• intervalData - should be true for "Net billing" process.
				• intervalDataDetailed - possible values:
				○ true - if needed generation graph detailed with power plants.

				○ false - if needed common generation graph without details. • intervalDataRecalculation - in this step should be true because we need to renew currently existing graph before extracting it. • objectNumbers - objectNumber from step 4.

6	receive order ID	Step 5 will return order ID.		**Response body**
				
				{
				"orderId": 1000003
				}

7	call for "Net billing" prosumer graph	After successful "Net billing" prosumer graph generation. Graph should be read page by page. **Note**: • for more details on how to work with orders look in Async. • for more details on how to use pagination look in Sync.	GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl Where: • {orderId} - Order ID from step 6. • header: first = 0 • header: count: 10 000 Recommended page size 10 000 is the max size of order page size to get whole report with one request. Order will contain graph for the one object only.	
8	receive "Net billing" prosumer graph	Receive prosumer graphs. **Note:** • For recalculated data usage type will be B - Billing. • After recalculation newer graph version will be captured in DH side and history change order will not show any changes for recalculated object. • It's up to supplier how to use recalculated data. ○ To create dummy adjusting bill from recalculated data		**Response body** [{ "consumptionCategories": [{ "consumptionCategory": "P-", "consumptions": [{ "amount": 45,

		and check if it matches old bill or not. ○ To compare old graph version with recalculated one and only then create adjusting bill. ○ Also supplier decides to create adjusting bills or not for customers which already left supplier.		"consumptionTime": "2024-05-10T18:00:000", "graphVersion": "2024-05-10T18:00:00.000", "usageType": "B", "valueType": "VAL" }], "powerPlantObjectNumber": "45654654", "powerPlantType": "A" }, ...], "objectBsId": 0, "objectNumber": "4565657", ... }]

7 DataHub Gatewy API documentation

This section of the document provides detailed information about the DataHub Application Programming Interface (DH API), including descriptions of API methods, the structure of request and response JSON data models, data validation rules, error handling procedures, and other related topics.

DH returns standard HTTP response codes:

HTTP response codes	Reason	Description
200	OK	The request has succeeded.
201	Created	The request was successful, and a new resource has been created.
204	No content	No data found according to the given parameters.
400	Bad Request	Request error. The HTTP response body provides a list of errors in JSON format.
401	Unauthorized	An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.
403	Forbidden	According to the access control policy, the current user does not have access to perform the requested action.
404	Not Found	Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.

When submitting requests that do not adhere to the described rules, a JSON error response will be returned in the following format:

Error response

```
{
   "errorMessages": [
      {
         "code": integer,
         "text": "string"
      }
   ]
}
```

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Mandatory	Description
code	integer	Y	Error code.
text	string	Y	Error message.

7.1 Order controller

7.1.1 POST /gateway/public-supplier/order/list

URL	POST /gateway/public-supplier/order/list
Description	Method will return list of the orders.
Parameter	**URL parameters:** • **first** - the index of the report line, which must be the first in the return list (starting from 0). Optional. The default value is 0. • **count** - the number of order's rows in the return list. Optional. The default value is 30. If no count value is given, the default value count will be 30. • **sort** – ASC, DSC sorting: • By default, the orders list must be sorted by the orderId.
Header	After decrypting the public supplier authentication key, the public supplier ID is used to select the data.
JSON request	{ "orderId": "integer", "orderTypes": ["string"], "submittedDateFrom": "dateTime", "submittedDateTo": "dateTime", "dateFrom": "date", "dateTo": "date", "latestStatuses": ["string"],

<table>
<tr><td></td><td colspan="3">"auto": "boolean",
"userNameSearch": "string",
"orderParametersSearch": "string"

}</td></tr>
<tr><td rowspan="6">HTTP Response code</td><td>HTTP status code</td><td>Reason</td><td>Description</td></tr>
<tr><td>200</td><td>OK</td><td>Request completed successfully.</td></tr>
<tr><td>204</td><td>No Content</td><td>No data found according to the given parameters.</td></tr>
<tr><td>400</td><td>Bad Request</td><td>Request error. The HTTP response body provides a list of errors in JSON format.</td></tr>
<tr><td>401</td><td>Unauthorized</td><td>An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.</td></tr>
<tr><td>403</td><td>Forbidden</td><td>According to the access control policy, the current user does not have access to perform the requested action.</td></tr>
<tr><td></td><td>404</td><td>Not Found</td><td>Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.</td></tr>
<tr><td>JSON Response</td><td colspan="3">[
{
"orderId": "integer",
"orderType": "string",
"submittedDate": "datetime",
"dateFrom": "date",
"dateTo": "date",
"orderParameters": "string",
"latestStatus": "string",
"statusDate": "datetime",
"expireDate": "datetime",
"auto": "boolean",
"userName": "string"
}
]</td></tr>
</table>

<table>
<tr><td>JSON error response</td><td colspan="5"><pre>{
  "errorMessages": [
    {
      "code": "integer",
      "text": "string"
    }
  ]
}</pre></td></tr>
<tr><td rowspan="4">Rules</td><td>No.</td><td>Rule description</td><td>Error code</td><td>Error message</td><td>Attributes</td></tr>
<tr><td>1.</td><td>If an attribute has defined possible values, the value index can be specified by specifying the value of the attribute in the request. Indices of all possible values start from 0.</td><td>-</td><td>-</td><td>All attributes with specified values.</td></tr>
<tr><td>2.</td><td>The date from cannot be later than the date to but can be equal.</td><td>1002</td><td>Date from cannot be later than date to.</td><td>dateFrom, dateTo, submittedDateFrom, submittedDateTo</td></tr>
<tr><td>3.</td><td>Submitted date cannot be later than the current date but can be equal.</td><td>1010</td><td>Submitted date cannot be later than the current date.</td><td>SubmittedDateFrom, SubmittedDateTo</td></tr>
</table>

7.1.1.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description
orderId	integer	not required	The report ordering primary surrogate key.
orderTypes	string	not required	The short name of the order type. Possible meanings: • data-hr-15min-obj-lvl - Automated quantities at the object level

			• data-hr-15min-history-changes - Net billing accounting scheme changes of interval data • balance-data - Balance data report • balance-by-generation-type – Balance by generation type report • balance-data-by-contract-type – Balance data by contract type
submittedDateFrom	datetime	not required	Order’s submission date from.
submittedDateTo	datetime	not required	Order’s submission date to.
dateFrom	date	not required	The beginning of the reporting period: • The format: YYYY-MM-DD.
dateTo	date	not required	The end of the reporting period: • The format: YYYY-MM-DD.
latestStatuses	[string]	not required	The status of the order. Possible meanings: • IV – Completed. • V – In progress. • P – Submitted. • K – Error. More than one type can be submitted.
auto	boolean	not required	Indication that the order was ordered automatically.
userNameSearch	string (240)	not required	The user who ordered the order.
orderParametersSearch	string	not required	The order parameters.

7.1.1.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description

orderId	integer	required	The report ordering primary surrogate key.
orderType	string (100)	required	The short name of the order type. Possible meanings: • data-hr-15min-obj-lvl - Automated quantities at the object level • data-hr-15min-history-changes - Net billing accounting scheme changes of interval data • balance-data - Balance data report • balance-by-generation-type – Balance by generation type report • balance-data-by-contract-type – Balance data by contract type
submittedDate	datetime	required	The date of the order submission.
dateFrom	date	required	The beginning of the reporting period: • The format: YYYY-MM-DD.
dateTo	date	required	The end of the reporting period: • The format: YYYY-MM-DD.
orderParameters	string (4000)	required	The search parameters by which the data in the ordered order was filtered.
latestStatus	string (20)	required	The current status of the order.
statusDate	dateTime	required	The latest status date.
expireDate	dateTime	required	Date of validity of the order. • The ordered report with **status = Completed** by default, is available only for 24 hours.
auto	boolean	required	Indication that the report order was ordered automatically.
userName	string (240)	required	The user who ordered the order.

7.1.1.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.2 GET /gateway/public-supplier/order/{orderId}/count

URL	GET /gateway/public-supplier/order/{orderId}/count		
Description	Method which will return count (number), how many items public supplier will get in ordered report (reports could have more than 1 item, so it is List). It should be used when public supplier needs to split data in few portions. This response should be used in reports' GET methods request, where public supplier can provide method parameters information.		
Parameters	**URL parameters:** • **orderId** – order identification number. Required.		
Header	After decrypting the public supplier authentication key, the public supplier ID is used to select the data.		
JSON request	GET request does not have the BODY part.		
Response	**HTTP status code**	**Reason**	**Description**
	200	OK	Request completed successfully.
	204	No content	No data found according to the given parameters.
	400	Bad Request	Request error. The HTTP response body provides a list of errors in JSON format.

	401	Unauthorized	An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.		
	403	Forbidden	According to the access control policy, the current user does not have access to perform the requested action.		
	404	Not Found	Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.		

JSON Response	<pre>{
	"count": "integer"
	}</pre>				

JSON error response	<pre>{
	"errorMessages": [
	{
	"code": "integer",
	"text": "string"
	}
	]
	}</pre>				

Rules	**No.**	**Rule description**	**Error code**	**Error message**	**Attributes**
	1.	Counts report list items and return SUM of all report lines.	-	-	-
	2.	The order status must be **Completed**.	2010	Invalid report order status.	orderId
	3.	Report order doesn't exist in the system.	2016	According to the submitted order number: **[orderId]**, the order does not exist.	orderId
	4.	Invalid method selected for report data or incorrect parameter.	2017	Invalid method selected or parameter specified incorrectly. According to the submitted order number: [**orderId**] report type is: [**orderType**].	orderId, orderType

	5.	No data found based on the search parameters submitted in the POST method.	2018	There is no data for the selected search parameters, the response is empty.	orderId

7.1.2.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description

7.1.2.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
count	integer	required	Number of rows, objects, accounts, depending on the selected report.

7.1.2.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.3 POST /gateway/public-supplier/order/data-hr-15min-obj-lvl

<table>
<tr><td>URL</td><td colspan="3">POST /gateway/public-supplier/order/data-hr-15min-obj-lvl</td></tr>
<tr><td>Description</td><td colspan="3">The method is designed for ordering data for “Automated quantities at the object level”. Using this method public supplier can only order the data of objects in public supply.</td></tr>
<tr><td>Parameter</td><td colspan="3">URL parameters:

The JSON data is contained in the HTTP request (BODY) (see JSON structure, below).</td></tr>
<tr><td>Header</td><td colspan="3">After decrypting the public supplier authentication key, the public supplier ID is used to select the data.</td></tr>
<tr><td>JSON request</td><td colspan="3">{
"dateFrom": "date",
"dateTo": "date",
"consumptionCategories": [
"string",
"string"
],
"objectNumbers": [
"string",
"string"
],
"interval": "string",
"netBilling": {
"intervalData": "boolean",
"intervalDataRecalculation": "boolean",
"intervalDataDetailed": "boolean"
}
}</td></tr>
<tr><td>HTTP Response code</td><td>HTTP status code</td><td>Reason</td><td>Description</td></tr>
<tr><td></td><td>201</td><td>Created</td><td>Request completed successfully.</td></tr>
</table>

	400	Bad Request	Request error. The HTTP response body provides a list of errors in JSON format.		
	401	Unauthorized	An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.		
	403	Forbidden	According to the access control policy, the current user does not have access to perform the requested action.		
	404	Not Found	Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.		

JSON response	<pre>{
	"orderId": "integer"
	}</pre>				

JSON error response	<pre>{
	"errorMessages": [
	{
	"code": "integer",
	"text": "string"
	}
	]
	}</pre>				

Rules	**No.**	**Rule description**	**Error code**	**Error message**	**Attributes**
	1.	If an attribute has defined possible values, the value index can be specified by specifying the value of the attribute in the request. Indices of all possible values start from 0.	-	-	All attributes with specified values.
	2.	The date from cannot be later than the date to but can be equal.	1002	Date from cannot be later than date to.	dateFrom, dateTo
	3.	The date from and / or date to cannot be later than the current date but can be equal.	1008	Date from and / or date to cannot be later than the current date.	dateFrom, dateTo
	4.	Object meter must be automated.	2007	The submitted object number: **[objectNumber (if there is more**	objectNumbers

				than one object, objects must be separated by the semicolon)], was not found or the meter of object is not automated.	
	5.	Data cannot be older than 36 months old.	2012	Date from cannot be older than 36 months old.	dateFrom
	6.	Report can be ordered maximum for 12 months.	2013	The report can only be ordered for 12 months or less.	dateFrom, dateTo
	7.	A maximum of 500 objects can be submitted in a report order.	2021	A maximum of 500 objects can be submitted in a report order.	objectNumbers
	8.	If objectNumbers [null], then the report can be ordered for a maximum of 1 month period.	2023	The report without specifying the objects can only be ordered for 1 month or less.	objectNumbers, dateFrom, dateTo
	9.	Parameters below can only be specified, if netBilling intervalData=TRUE: • intervalDataRecalculation • intervalDataDetailed **Note.** Without specifying the latter parameters, i.e. specifying NULL will treat them as FALSE.	2026	Recalculation of generation and consumption and an option to choose the type of power plant data view is only possible if the order is submitted for the object, which has "Net billing" accounting scheme.	netBilling, intervalData, intervalDataRecalculation, intervalDataDetailed
	10.	If netBilling intervaleDataRecalculation=TRUE, then recalculation can be initiated only for past periods.	2027	Recalculation of generation and consumption for object which has "Net billing" accounting scheme can be only initiated for past periods.	netBilling, intervalDataRecalculation, dateFrom, dateTo
	11.	The meaning of the **[ObjectNumber]** cannot be repeated.	2028	The object: [**objectNumber (if there is more than one object, objects must be separated by the semicolon)**] is repeating.	objectNumber
	12.	If **netBilling.intervalData=true** and **netBilling .intervalDataRecalculation=true**, then recalculation can be initiated for the previous month only possible from the 2nd working day	2030	Recalculation of generation and consumption for object which has "Net billing" accounting scheme is not possible for the previous accounting	netBilling, intervalData**,** intervalDataRecalculation

		at 9 o'clock of the current month (date and hour configurated parameter). Examples: • If current date and time **2024-04-03 8 hours (it is the 2nd working day 8 hour)**, then recalculation of generation and consumption for object which has "Net billing" accounting is not possible for 2024-03 accounting period. • If current date and time **2024-04-03 10 hours (it is the 2nd working day 10 hour)**, then recalculation of generation and consumption for object which has "Net billing" accounting scheme is possible for 2024-03 accounting period.		period (**previous accounting period** [**YYYY-MM**]).	
	13.	If netBilling **intervalData=TRUE**, i**ntervalDataRecalculation=TRUE**, then recalculation of generation and consumption for object which has "Net billing" accounting scheme can be initiated for only 1 object in one request and only for 1 reporting period. **Note**. If the calculation period is not a full month (for example one day), then the entire month is calculated, and the data is provided only for the period that was specified during the order. Examples: • Recalculation period 2024-02-01 - 2024-02-29 (complies with requirements); • Recalculation period 2024-02-15 - 2024-03-15 (does not complies requirements, because specified 2 accounting months); • Recalculation period 2024-02-15- 2024-02-15 (complies with requirements).	2032	Recalculation of generation and consumption for object which has "Net billing" accounting scheme can be initiated only for 1 object and only for 1 accounting period.	netBilling, intervalData, intervalDaterecalculation

7.1.3.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description
dateFrom	date	required	The beginning of the reporting period: • The format: YYYY-MM-DD.
dateTo	date	required	The end of the reporting period: • The format: YYYY-MM-DD.
consumptionCategory	[string (20)]	required	The consumption category. Possible meanings: • P+ (active P+ electricity). • P- (active P- electricity). • Q+ (reactive Q+ electricity). • Q- (reactive Q+ electricity).
objectNumbers	[string]	required	Object numbers.
interval	string	required	Consumption interval. Possible meanings: • HOUR • QUARTER
netBilling: {}			
intervalData	boolean	not required	Indication that the object is in "Net billing" accounting scheme. Possible meanings: • TRUE • FALSE • NULL Default value is NULL. NULL is treated as False.
intervalDataRecalculation	boolean	not required	Indication for the recalculation of objects in the "Net billing" accounting scheme. Possible meanings: • TRUE • FALSE

			• NULL Default value is NULL. NULL is treated as False.
intervalDataDetailed	boolean	not required	Indication of whether object in "Net billing" accounting scheme detailed information should be retrieved. Possible meanings: • TRUE – a detailed view will be returned (the consumption object and all its power plant objects); • FALSE – a aggregated view will be returned (the consumption object without power plant objects); • NULL. Default value is NULL. NULL is treated as False.

7.1.3.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
orderId	integer	required	The report ordering primary surrogate key.

7.1.3.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.4 GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl

<table>
<tr><td>URL</td><td colspan="3">GET /gateway/public-supplier/order/{orderId}/data-hr-15min-obj-lvl?first={integer}&count={integer}</td></tr>
<tr><td>Description</td><td colspan="3">The method for receive the order report "Automated quantities at the object level".</td></tr>
<tr><td>Parameter</td><td colspan="3">URL parameters:

• orderId – order identification number.
• first - the index of the object, which must be the first in the return list (starting from 0). Optional. The default value is 0.
• count - the number of objects in the return list. Optional. The default value is 10000. If no count value is given, the default value count will be 10000.</td></tr>
<tr><td>Header</td><td colspan="3">After decrypting the public supplier authentication key, the public supplier ID is used to select the data.</td></tr>
<tr><td>JSON request</td><td colspan="3"></td></tr>
<tr><td rowspan="7">HTTP Response code</td><td>HTTP status code</td><td>Reason</td><td>Description</td></tr>
<tr><td>200</td><td>OK</td><td>Request completed successfully.</td></tr>
<tr><td>204</td><td>No content</td><td>No data found according to the given parameters.</td></tr>
<tr><td>400</td><td>Bad Request</td><td>Request error. The HTTP response body provides a list of errors in JSON format.</td></tr>
<tr><td>401</td><td>Unauthorized</td><td>An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.</td></tr>
<tr><td>403</td><td>Forbidden</td><td>According to the access control policy, the current user does not have access to perform the requested action.</td></tr>
<tr><td>404</td><td>Not Found</td><td>Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.</td></tr>
<tr><td>JSON response</td><td colspan="3">{
"personCode": "string",
"personName": "string",
"personSurname": "string",</td></tr>
</table>

```
  "objectId": "integer",
  "objectNumber": "string",
  "consumptionCategories": [
    {
      "consumptionCategory": "string",
      "powerPlantObjectNumber": "string",
      "powerPlantType": "string",
      "consumptions": [
        {
          "consumptionTime": "datetime",
          "amount": "number",
          "valueType": "string",
          "usageType": "string",
          "graphVersion": "dateTime"
        }
      ]
    }
  ]
}
```


JSON error response


```
{
  "errorMessages": [
    {
      "code": "integer",
      "text": "string"
    }
  ]
}
```


Rules

No.	Rule description	Error code	Error message	Attributes
1.	The order status must be Completed**.**	2010	Invalid report order status.	orderId
2.	According to the submitted order number: **[orderId]**, the order does not exist.	2016	Report order doesn't exist in the system.	orderId
3.	Invalid method selected or parameter specified incorrectly. According to the submitted order number: [**orderId**] report type is: [**orderType**].	2017	Invalid method selected for report data or incorrect parameter.	orderId, orderType

	4.	No data found based on the search parameters submitted in the POST method.	2018	There is no data for the selected search parameters, the response is empty.	orderId
	5.	The number of objects in the return list must be less than or equal to 10000.	2022	The number of objects in the return list must be less than or equal to [**10000**].	count

7.1.4.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description

7.1.4.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
personCode	string (20)	required	Person / company code.
personName	string (200)	required	Person / company name.
personSurname	string (50)	required	Person surname.
objectBsId	integer	required	Object Id.
objectNumber	string (20)	required	Object number.

consumptionCategories: []			
consumptionCategory	string (2)	required	Consumption category. Possible meanings: • P+ (active P+ electricity); • P- (active P- electricity); • Q+ (reactive Q+ electricity); • Q- (reactive Q+ electricity).
powerPlantObjectNumber	string(20)	not required	Power plant object number, which has "Net billing" accounting scheme. Note: Filled in if attributes were selected when ordering the report: intervalData: **true** AND intervalDataDetailed: **true**.
powerPlantType	string	not required	Type of the power plant. Possible meanings: • A – Waste fuel. • B – Biomass. • H – Hydroelectric. • K – Other. • S – Solar. • T – TEC. • V – Wind. • P – Storage device. • I – Fossil. • D – Biogas. • R – Hybrid generation. Note: Filled in if attributes were selected when ordering the report: intervalData: **true** AND intervalDataDetailed: **true**.
consumptions: []			
consumptionTime	datetime with timeZone	required	Consumption time.

amount	number(6)	required	Consumption amount in kWh/kVArh.
valueType	string (3)	required	Consumption value type. Possible meanings: • EST – estimated. • VAL – validated.
usageType	string(1)	not required	Reading usage type (only for object, which has "Net billing" accounting scheme). Possible field values are: • B – Billing; • D – Daily Note: Filled in if attributes were selected when ordering the report: intervalData: **true**.
graphVersion	dateTime	not required	Calculated version of the "Net billing" accounting scheme graph. Note: Filled in if attributes were selected when ordering the report: intervalData: **true**.

7.1.4.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.5 POST /gateway/public-supplier/order/data-hr-15min-history-changes

<table>
<tr><td>URL</td><td colspan="3">POST /gateway/public-supplier/order/data-hr-15min-history-changes</td></tr>
<tr><td>Description</td><td colspan="3">The method is intended for ordering "Net billing accounting scheme changes of interval data" report.</td></tr>
<tr><td>Parameter</td><td colspan="3">URL parameters:

The JSON data is contained in the HTTP request (BODY) (see JSON structure, below).</td></tr>
<tr><td>Header</td><td colspan="3">After decrypting the public supplier authentication key, the public supplier ID is used to select the data.</td></tr>
<tr><td>JSON request</td><td colspan="3">{
"dateFrom": "date",
"objectNumbers": [
"string"
]
}</td></tr>
<tr><td>HTTP Response code</td><td>HTTP status code</td><td>Reason</td><td>Description</td></tr>
<tr><td></td><td>201</td><td>Created</td><td>Request completed successfully.</td></tr>
<tr><td></td><td>400</td><td>Bad Request</td><td>Request error. The HTTP response body provides a list of errors in JSON format.</td></tr>
<tr><td></td><td>401</td><td>Unauthorized</td><td>An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.</td></tr>
<tr><td></td><td>403</td><td>Forbidden</td><td>According to the access control policy, the current user does not have access to perform the requested action.</td></tr>
<tr><td></td><td>404</td><td>Not Found</td><td>Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.</td></tr>
<tr><td>JSON response</td><td colspan="3">{
"orderId": "integer"
}</td></tr>
<tr><td>JSON error response</td><td colspan="3">{
"errorMessages": [
{</td></tr>
</table>

<table>
<tr><td></td><td colspan="5">"code": "integer",
"text": "string"
}
]
}</td></tr>
<tr><td>Rules</td><td>No.</td><td>Rule description</td><td>Error code</td><td>Error message</td><td>Attributes</td></tr>
<tr><td></td><td>1.</td><td>The date from cannot be later than the current date but can be equal.</td><td>1008</td><td>The date from and / or date to cannot be later than the current date.</td><td>dateFrom</td></tr>
<tr><td></td><td>2.</td><td>Object meter must be automated.</td><td>2007</td><td>The submitted object number: [objectNumber (if there is more than one object, objects must be separated by the semicolon)], was not found or the meter of object is not automated.</td><td>objectNumber</td></tr>
<tr><td></td><td>3.</td><td>A maximum of [configurated parameter] objects can be submitted in a report order.</td><td>2021</td><td>A maximum of {500} objects can be specified.</td><td>objectNumbers</td></tr>
<tr><td></td><td>4.</td><td>The meaning of the [ObjectNumber] notification cannot be repeated.</td><td>2028</td><td>The object: [objectNumber (if there is more than one object, objects must be separated by the semicolon)] is repeating.</td><td>objectNumbers</td></tr>
<tr><td></td><td>5.</td><td>Data for selected report can by ordered if report is not locked.</td><td>2031</td><td>Data is not currently available for the selected report.</td><td>dateFrom</td></tr>
<tr><td></td><td>6.</td><td>Report can be ordered maximum for 3 accounting months, excluding the current month.

Example:

• If current date 2024-06-28 (month 06 is excluded and 3 full calendar months are counted in the past 05, 04, 03) → maximum order data from value must be 2024-03-01.</td><td>2033</td><td>Report can be ordered maximum for 3 previous accounting months.</td><td>dateFrom</td></tr>
</table>

		• If date from of the report is selected less than 2024-03-01 → the system displays error message.			
	7.	By default, order dateTo is set to the current date.			

7.1.5.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description
dateFrom	date	required	The date from which we want to obtain information on possible recalculations of the objects. • The format: YYYY-MM-DD.
objectNumbers	["string"]	not required	Objects for which we want to check if there are possible recalculations.

7.1.5.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
orderId	integer	required	The report ordering primary surrogate key.

7.1.5.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.6 GET /gateway/public-supplier/order/{orderId}/data-hr-15min-history-changes

<table>
<tr><td>URL</td><td colspan="3">GET /gateway/public-supplier/order/{orderId}/data-hr-15min-history-changes</td></tr>
<tr><td>Description</td><td colspan="3">The method is designed to obtain ''Net billing accounting scheme changes of interval data'' report.</td></tr>
<tr><td>Parameters</td><td colspan="3">URL parameters:

• orderId – order identification number.
• first - the index of the object, which must be the first in the return list (starting from 0). Optional. The default value is 0.
• count - the number of objects in the return list. Optional. The default value is 10000. If no count value is given, the default value count will be 10000.</td></tr>
<tr><td>Header</td><td colspan="3">After decrypting the public supplier authentication key, the public supplier ID is used to select the data.</td></tr>
<tr><td>JSON request</td><td colspan="3"></td></tr>
<tr><td rowspan="6">Response HTTP code</td><td>HTTP status code</td><td>Reason</td><td>Description</td></tr>
<tr><td>200</td><td>OK</td><td>Request completed successfully.</td></tr>
<tr><td>400</td><td>Bad Request</td><td>Request error. The HTTP response body provides a list of errors in JSON format. (Error response of the JSON structure below).</td></tr>
<tr><td>401</td><td>Unauthorized</td><td>An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.</td></tr>
<tr><td>403</td><td>Forbidden</td><td>According to the access control policy, the current user does not have access to perform the requested action.</td></tr>
<tr><td>404</td><td>Not Found</td><td>Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.</td></tr>
<tr><td>JSON response</td><td colspan="3">[
{
"personCode": "string",</td></tr>
</table>

```
        "personName": "string",
        "personSurname": "string",
        "objectNumber": "string",
        "periodsWithChanges": [
          {
            "billingPeriod": "date",
            "reasons": [
              "string",
              "string"
            ]
          }
        ]
    }
]
```


JSON error response


```
{
    "errorMessages": [
        {
            "code": "integer",
            "text": "string"
        }
    ]
}
```


Rules

No.	Rule description	Error code	Error message	Attributes
1.	The order status must be **Completed**.	2010	Invalid report order status.	orderId
2.	Report order doesn't exist in the system.	2016	According to the submitted order number: **[orderId]**, the order does not exist.	orderId
3.	Invalid method selected for report data or incorrect parameter.	2017	Invalid method selected or parameter specified incorrectly. According to the submitted order number: **[orderId]** report type is: **[orderType].**	orderId, orderType

	4.	No data found based on the search parameters submitted in the POST method.	2018	There is no data for the selected search parameters, the response is empty.	orderId
	5.	The number of objects in the return list must be less than or equal to 10000.	2022	The number of objects in the return list must be less than or equal to [**10000**].	count

7.1.6.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description

7.1.6.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
personCode	string	required	Person / company code.
personName	string	required	Person name / company name.
personSurname	string	not required	Person surname.
objectNumber	string (20)	required	Object number.
periodWithChanges: []			

billingPeriod	date	required	Accounting period which has possible recalculations for consumption object. • The format: YYY-MM
reasons	["string"]	required	Reasons for change: • GENERATION_CHANGE (examples: power plant power changes, meter reading changes, permissible power generation changes) • OWNER_CHANGE • SUPPLIER_CHANGE

7.1.6.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.7 POST /gateway/public-supplier/order/balance-data

URL	POST /gateway/public-supplier/order/balance-data
Description	The method is intended for ordering balance data report.
Parameter	**URL parameters:** The JSON data is contained in the HTTP request (BODY) (*see JSON structure, below*).
Header	After decrypting the public supplier authentication key, the public supplier ID is used to select the data.
JSON request	{ "dateFrom": "date",

<table>
<tr><td></td><td colspan="5">"dateTo": "date",
"interval": "string"
}</td></tr>
<tr><td rowspan="6">HTTP Response code</td><td>HTTP status code</td><td colspan="2">Reason</td><td colspan="2">Description</td></tr>
<tr><td>201</td><td colspan="2">Created</td><td colspan="2">Request is completed successfully.</td></tr>
<tr><td>400</td><td colspan="2">Bad Request</td><td colspan="2">Request error. The HTTP response body provides a list of errors in JSON format.</td></tr>
<tr><td>401</td><td colspan="2">Unauthorized</td><td colspan="2">An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.</td></tr>
<tr><td>403</td><td colspan="2">Forbidden</td><td colspan="2">According to the access control policy, the current user does not have access to perform the requested action.</td></tr>
<tr><td>404</td><td colspan="2">Not Found</td><td colspan="2">Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.</td></tr>
<tr><td>JSON response</td><td colspan="5">{
"orderId": "integer"
}</td></tr>
<tr><td>JSON error response</td><td colspan="5">{
"errorMessages": [
{
"code": "integer",
"text": "string"
}
]
}</td></tr>
<tr><td rowspan="2">Rules</td><td>No.</td><td>Rule description</td><td>Error code</td><td>Error message</td><td>Attributes</td></tr>
<tr><td>1.</td><td>The date from cannot be later than the date to. Equal can be.</td><td>1002</td><td>Date from cannot be later than date to.</td><td>dateFrom, dateTo</td></tr>
</table>

	2.	The date from and / or date to cannot be later than the current date but can be equal.	1008	Date from and / or date to cannot be later than the current date.	dateFrom, dateTo
	3.	Date cannot be older than 36 months old.	2012	Date from cannot be older than 36 months old.	dateFrom
	4.	The period date from and date to must be less than date from configurable parameter.	2015	Data is not currently available for the selected reporting period.	dateFrom, dateTo
	5.	Report can be ordered maximum for 1 accounting month.	2024	The report can only be ordered for 1 accounting month or less.	dateFrom, dateTo

7.1.7.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description
dateFrom	date	required	The beginning date (inclusive) of the reporting period: • The format: YYYY-MM-DD;
dateTo	date	required	The end date of the reporting period: • The format: YYYY-MM-DD;
interval	date	required	Consumption interval. Possible meanings: • HOUR • QUARTER

7.1.7.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
orderId	integer	required	The report ordering primary surrogate key.

7.1.7.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.8 GET /gateway/public-supplier/order/{orderId}/balance-data

URL	GET /gateway/public-supplier/order/{orderId}/balance-data
Description	The method is used to obtain the report "Balance data"
Parameters	**URL parameters:** • **orderId** – order identification number. • **first** - the index of the object, which must be the first in the return list (starting from 0). Optional. The default value is 0. • **count** - the number of objects in the return list. Optional. The default value is 10000. If no count value is given, the default value count will be 10000.
Header	After decrypting the public supplier authentication key, the public supplier ID is used to select the data.
JSON request	

Response HTTP code	HTTP status code	Reason	Description
	200	OK	Request is completed successfully.
	400	Bad Request	Request error. The HTTP response body provides a list of errors in JSON format. (*Error response of the JSON structure below*).
	401	Unauthorized	An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.
	403	Forbidden	According to the access control policy, the current user does not have access to perform the requested action.
	404	Not Found	Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.

JSON response	<pre>{
	  "timeSeriesData": [
	  {
	  "intervalDateTime": "dateTime",
	  "valueOfGeneration": "double(10.3)",
	  "valueOfConsumption": "double(10.3)"
	  }
	  ]
	}</pre>		

JSON error response	<pre>{
	  "errorMessages": [
	  {
	  "code": "integer",
	  "text": "string"
	  }
	  ]
	}</pre>		

Rules	No.	Rule description	Error code	Error message	Attributes
	1.	The order status must be **Completed**.	2010	Invalid report order status.	orderId
	2.	Report order doesn't exist in the system.	2016	According to the submitted order number: **[orderId]**, the order does not exist.	orderId
	3.	Invalid method selected for report data or incorrect parameter.	2017	Invalid method selected or parameter specified incorrectly. According to the submitted order number: **[orderId]** report type is: **[orderType].**	orderId, orderType
	4.	No data found based on the search parameters submitted in the POST method.	2018	There is no data for the selected search parameters, the response is empty.	orderId
	5.	Data must be sorted by date ascending	-	-	-

7.1.8.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description

7.1.8.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
timeSeriesData: []			
intervalDateTime	dateTime with timeZone	required	Date interval.
valueOfGeneration	double (10.3)	required	Total generated electricity in MWh.
valueOfConsumption	double (10.3)	required	Total consumed electricity in MWh.

7.1.8.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.9 POST /gateway/public-supplier/order/balance-by-generation-type

URL	POST /gateway/public-supplier/order/ balance-by-generation-type		
Description	The method is intended for ordering "Balance by generation type" report.		
Parameter	**URL parameters:** The JSON data is contained in the HTTP request (BODY) (*see JSON structure, below*).		
Header	After decrypting the public supplier authentication key, the public supplier ID is used to select the data.		

JSON request	{
	"generationType": [
	"string",
	"string"
	],
	"generationCategory": [
	"string",
	"string"
	],
	"dateFrom": "date",
	"dateTo": "date",
	"interval": "string"
	}		

HTTP Response code	**HTTP status code**	**Reason**	**Description**
	201	Created	Request is completed successfully.
	400	Bad Request	Request error. The HTTP response body provides a list of errors in JSON format.
	401	Unauthorized	An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.

	403	Forbidden	According to the access control policy, the current user does not have access to perform the requested action.
	404	Not Found	Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.

JSON response

```
{
 "orderId": "integer"
}
```

JSON error response

```
{
   "errorMessages": [
      {
         "code": "integer",
         "text": "string"
      }
   ]
}
```

Rules

No.	Rule description	Error code	Error message	Attributes
1.	The date from cannot be later than the date to. Equal can be.	1002	Date from cannot be later than date to.	dateFrom, dateTo
2.	The date from and / or date to cannot be later than the current date but can be equal.	1008	Date from and / or date to cannot be later than the current date.	dateFrom, dateTo
3.	Date cannot be older than 36 months old.	2012	Date from cannot be older than 36 months old.	dateFrom
4.	The period date from and date to must be less than date from configurable parameter.	2015	Data is not currently available for the selected reporting period.	dateFrom, dateTo
5.	Report can be ordered maximum for 1 accounting month.	2024	The report can only be ordered for 1 accounting month or less.	dateFrom, dateTo

7.1.9.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description
generationType	string	not required	Generation type. Possible values: • A – Waste fuel. • B – Biomass generation. • H – Hydroelectric generation. • K – Other generation. • S – Solar generation. • T – TEC generation. • V – Wind generation. • P – Storage device. • I – Fossil. • D – Biogas. • R – Hybrid generation.
generationCategory	string	not required	Producer's category. Possible values: • PRODUCERS • PROSUMERS • UNALLOCATED • REMOTE-PROSUMERS
dateFrom	date	required	The beginning date (inclusive) of the reporting period: • The format: YYYY-MM-DD;
dateTo	date	required	The end date of the reporting period: • The format: YYYY-MM-DD;
interval	date	required	Consumption interval. Possible meanings: • HOUR • QUARTER

7.1.9.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
orderId	integer	required	The report ordering primary surrogate key.

7.1.9.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.10 GET /gateway/public-supplier/order/{orderId}/balance-by-generation-type

URL	GET /gateway/public-supplier/order/{orderId}/balance-by-generation-type
Description	The method is used to obtain the report "Balance by generation type"
Parameters	**URL parameters:** • **orderId** – order identification number. • **first** - the index of the object, which must be the first in the return list (starting from 0). Optional. The default value is 0. • **count** - the number of objects in the return list. Optional. The default value is 10000. If no count value is given, the default value count will be 10000.
Header	After decrypting the public supplier authentication key, the public supplier ID is used to select the data.

JSON request			
Response HTTP code	**HTTP status code**	**Reason**	**Description**
	200	OK	Request is completed successfully.
	400	Bad Request	Request error. The HTTP response body provides a list of errors in JSON format. (*Error response of the JSON structure below*).
	401	Unauthorized	An attempt was made to connect to a non-public method that requires authentication, but no user credentials were provided.
	403	Forbidden	According to the access control policy, the current user does not have access to perform the requested action.
	404	Not Found	Either there is no API method associated with the request URL path, or the request contains one or more parameters that did not return the data.
JSON response			


```
[
   {
      "generationType": "string",
      "timeSeriesData": [
         {
            "intervalDateTime": "dateTime",
            "generationCategories": [
               {
                  "generationCategory": "string",
                  "valueOfGeneration": "double(10.3)"
               }
            ]
         }
      ]
   }
]
```

JSON error response	<pre>{
	"errorMessages": [
	{
	"code": "integer",
	"text": "string"
	}
	]
	}</pre>				

Rules	**No.**	**Rule description**	**Error code**	**Error message**	**Attributes**
	1.	The order status must be **Completed**.	2010	Invalid report order status.	orderId
	2.	Report order doesn't exist in the system.	2016	According to the submitted order number: **[orderId]**, the order does not exist.	orderId
	3.	Invalid method selected for report data or incorrect parameter.	2017	Invalid method selected or parameter specified incorrectly. According to the submitted order number: **[orderId]** report type is: **[orderType].**	orderId, orderType
	4.	No data found based on the search parameters submitted in the POST method.	2018	There is no data for the selected search parameters, the response is empty.	orderId
	5.	Data must be sorted by date ascending	-	-	-

7.1.10.1 JSON Request structure

The table below describes the structure of the JSON request:

Attribute	Type	Obligation	Description

7.1.10.2 JSON Response structure

The table below describes the structure of the JSON response:

Attribute	Type	Obligation	Description
generationType	string	required	Generation type. Possible values: • A – Waste fuel. • B – Biomass generation. • H – Hydroelectric generation. • K – Other generation. • S – Solar generation. • T – TEC generation. • V – Wind generation. • P – Storage device. • I – Fossil. • D – Biogas. • R – Hybrid generation.
timeSeriesData: []			
intervalDateTime	dateTime with timeZone	required	Date interval.
generationCategories: []			
generationCategory	string	required	Producer’s category. Possible values: • PRODUCERS • PROSUMERS • UNALLOCATED

			• REMOTE-PROSUMERS
valueOfGeneration	double (10.3)	required	Total generated electricity in MWh.

7.1.10.3 Error Response Structure

The following table describes the JSON structure in the event of a response error:

Attribute	Type	Obligation	Description
code	integer	required	Error code.
text	string (4000)	required	Error message.

7.1.11 POST /gateway/public-supplier/order/balance-data-by-contract-type

URL	POST /gateway/public-supplier/order/balance-data-by-contract-type
Description	The method is intended for ordering "Balance data by contract type" report.
Parameter	No parameters.
Header	After decrypting the involved party's authentication key, the involved party ID is used to select the data.
HTTP response code	Standard HTTP response codes should be applied. The list of codes can be found at the following source: Standard HTTP response codes

JSON request	{
	"contractType": "string",
	"dateFrom": "string",
	"dateTo": "string",
	"interval": "string"
	}

JSON response	{

	"orderId": integer }
JSON error response	Example and description of JSON error response can be found at the following source: JSON error response

The table below describes the rules:

Rule description	Error code	Error message	Attributes
The date from cannot be later than the date to. Equal can be.	1002	Date from cannot be later than date to.	dateFrom, dateTo
The date from and / or date to cannot be later than the current date but can be equal.	1008	Date from and / or date to cannot be later than the current date.	dateFrom, dateTo
Date cannot be older than 36 months old.	2012	Date from cannot be older than 36 months old.	dateFrom
The period date from and date to must be less than date from configurable parameter.	2015	Data is not currently available for the selected reporting period.	dateFrom, dateTo
Report can be ordered maximum for 1 accounting month.	2024	The report can only be ordered for 1 accounting month or less.	dateFrom, dateTo

7.1.11.1 JSON request structure

The table below describes the structure of the JSON request:

Attribute	Type	Mandatory	Description
contractType	string	N	Contract type. Possible meanings: • SKMS • SBTS
dateFrom	string (date)	Y	The beginning date (inclusive) of the reporting period.

dateTo	string (date)	Y	The end date of the reporting period.
interval	string	Y	Consumption interval. Possible meanings: • HOUR • QUARTER

7.1.11.2 JSON response structure

The table below describes the structure of the JSON response:

Attribute	Type	Mandatory	Description
orderId	integer	Y	The report ordering primary surrogate key.

7.1.12 GET /gateway/public-supplier/order/{orderId}/balance-data-by-contract-type

URL	GET /gateway/public-supplier/order/{orderId}/balance-data-by-contract-type
Description	The method is used to obtain the report "Balance data by contract type".
Parameter	URL parameters: *orderId, first, count.*
Header	After decrypting the involved party's authentication key, the involved party ID is used to select the data.
HTTP response code	Standard HTTP response codes should be applied. The list of codes can be found at the following source: Standard HTTP response codes
JSON request	
JSON response	[{ "contractType": "string", "timeSeriesData": [{ "intervalDateTime": "string", "valueOfConsumption": number

	}] }]
JSON error response	Example and description of JSON error response can be found at the following source: JSON error response

The table below describes the rules:

Rule description	Error code	Error message	Attributes
The order status must be Completed.	2010	Invalid report order status.	orderId
Report order doesn't exist in the system.	2016	According to the submitted order number: *[orderId]*, the order does not exist.	orderId
Invalid method selected for report data or incorrect parameter.	2017	Invalid method selected or parameter specified incorrectly. According to the submitted order number: *[orderId]* report type is: *[orderType].*	orderId, orderType
No data found based on the search parameters submitted in the POST method.	2018	There is no data for the selected search parameters, the response is empty.	orderId
Data must be sorted by date ascending.			

7.1.12.1 Parameters

The table below describes the parameters:

Attribute	Type	Mandatory	Description
orderId *(path)*	integer	Y	Order identification number.

first *(query)*	integer	N	The index of the report line, which must be the first in the return list (starting from 0). The default value is 0.
count *(query)*	integer	N	The number of order’s rows in the return list. If no count value is given, the default value count will be 10 000.

7.1.12.2 JSON response structure

The table below describes the structure of the JSON response:

Attribute	Type	Mandatory	Description
contractType	string	Y	Contract type. Possible meanings: • SKMS • SBTS
timeSeriesData: []			
intervalDateTime	string (dateTime)	Y	Date interval (Date format example: 2025-01-01T00:00:00+02:00).
valueOfConsumption	number	Y	Total consumed electricity in MWh.

7.2 Object controller

7.2.1 POST /gateway/public-supplier/object/all/active/list

URL	POST /gateway/public-supplier/object/all/active/list
Description	The method is used to retrieve a list of objects based on selection criteria.
Parameter	URL parameters: *first, count, sort.*
Header	After decrypting the involved party's authentication key, the involved party ID is used to select the data.
HTTP response code	Standard HTTP response codes should be applied. The list of codes can be found at the following source: Standard HTTP response codes
JSON request	{ "personCode": "string", "consumerCode": "string", "objectNumber": "string", "meterNumber": "string", "objectDataConsentSign": boolean }
JSON response	[{ "personName": "string", "personSurname": "string", "personCode": "string", "consumerCode": "string", "objectNumber": "string", "metersAmount": integer, "autoMetersAmount": integer, "smartMeterInstallationDate": "string", "meters": [{ "meterNumber": "string", "meterAutomated": boolean, "automationSystem": "string", "meterScaleLength": integer, "scales": [{

```
                "scaleIdentifier": "string",
                "scaleProduct": "string"
            }
        ]
    }
],
"objectAddress": "string",
"contractType": "string",
"contractModel": "string",
"supplierType": "string",
"timeLimitedObjectValidTo": "string",
"tariffPlan": "string",
"tariffPlanChangeDate": "string",
"timeZone": "string",
"ownershipDocumentNumber": "string",
"supplyOwnership": "string",
"contractStart": "string",
"contractEnd": "string",
"accountingType": "string",
"objectFutureSuppliers": [
    {
        "supplierCode": "string",
        "supplierName": "string",
        "contractStart": "string",
        "contractEnd": "string"
    }
],
"usedPowerPlants": [
    {
        "powerPlantObjectNumber": "string",
        "powerPlantType": "string",
        "generatingObjectType": "string",
        "generatingPower": number,
        "powerPlantValidFrom": "string",
        "powerPlantValidTo": "string",
        "accountingScheme": "string",
        "accountingSchemeValidFrom": "string",
        "accountingSchemeValidTo": "string",
        "accountingSchemeChangeDate": "string",
        "payoffMethod": "string",
        "payoffMethodChangeDate": "string"
    }
],
"objectPowers": [
    {
        "powerType": "string",
```

<table>
<tr><td></td><td>
<pre>
            "power": number,
            "powerValidFrom": "string",
            "powerValidTo": "string"
        }
    ],
    "generatingObjectGroup": {
        "generatingGroup": integer,
        "generatingObjectPriorityGroup": integer
    },
    "objectLatestSupplyState": {
        "state": "string",
        "stateValidFrom": "string",
        "stateValidTo": "string"
    },
    "contact": {
        "mobPhoneNoNetwork": "string",
        "mobPhoneNo2Network": "string",
        "mobPhoneInvoice": "string",
        "phoneNoNetwork": "string",
        "emailNetwork": "string",
        "emailNetwork2": "string",
        "emailInvoice": "string"
    },
    "consumptionAverage": number,
    "consumptionAverageCalculationDate": "string",
    "consumptionAverageCalculationMonthsCount": integer,
    "powerPlantObjectType": "string"
  }
]
</pre>
</td></tr>
<tr><td>JSON error response</td><td>Example and description of JSON error response can be found at the following source: JSON error response</td></tr>
</table>

The table below describes the rules:

Rule description	Error code	Error message	Attributes
If an attribute has defined possible values, the value index can be specified by specifying the value of the attribute in the request. Indices of all possible values start from 0.			All attributes with specified values.
The request parameter "objectDataConsentSign" is true is required.	1020	It is mandatory to specify, that to obtain consent to see object information.	objectDataConsentSign
One or more request parameters are required.	1001	One or more request parameters are required.	personCode, consumerCode, objectNumber, meterNumber

7.2.1.1 Parameters

The table below describes the parameters:

Attribute	Type	Mandatory	Description
first (*query*)	integer	N	The index of the client object (starting from 0) that must be presented first in the return list. The default value is 0.
count (*query*)	integer	N	The number of client objects in the return list. The default value is 30.
sort (*query*)	string	N	Sorting order: ASC (ascending) or DESC (descending). The list entries can be sorted by any field in the list. By default, the list is sorted by the ConsumerCode field.

7.2.1.2 JSON request structure

The table below describes the structure of the JSON request:

Attribute	Type	Mandatory	Description
personCode	string (20)	N	Person / company code.
consumerCode	string (20)	N	Consumer code.
objectNumbers	string (20)	N	Object number. A maximum of 500 objects can be specified.
meterNumber	string (20)	N	Meter number.
objectDataConsentSign	boolean	N	Permission to view object information – indicates whether consent has been granted to access object details. Possible values: • True – consent granted. • False – consent not granted.

7.2.1.3 JSON response structure

The table below describes the structure of the JSON response:

Attribute	Type	Mandatory	Description
personName	string	Y	Name of the contract owner (individual or company).
personSurname	string	N	Surname of the contract owner or tenant.
personCode	string	N	Contract owner / tenant person or company code. If the subject is an individual, the personal code must be encrypted in the following format: [********] [last 3 digits of the personal code].

Attribute	Type	Mandatory	Description
consumerCode	string	Y	Consumer code of the contract owner or tenant.
objectNumber	string	Y	Object number.
meterAmount	integer	N	Total number of meters assigned to the object
autoMetersAmount	integer	N	Total number of automated meters assigned to the object.
smartMeterInstallationDate	string (date)	N	Installation date of the smart meter in the object
meters: []			
meterNumber	string	Y	Object's meter number
meterAutomated	boolean	Y	Flag indicating whether the meter is automated.
automationSystem	string	N	Automated Meter System. Possible values: • MDM – a new Meter Data Management system used with smart meters. • EMCOS – the existing EMCOS system used with automated meters.
meterScaleLenght	integer	Y	Maximum programmable number of scale marks installed in the meter.
scales: []			
scaleIdentifier	string	N	Internal scale identifier. Possible values: • VT • DD • DN • +QsumTS • +WsumT1 • +WsumT2 • +WsumT3

Attribute	Type	Mandatory	Description
			• +WsumT4 • -QsumTS • -WsumTS
scaleProduct	string	N	Internal scale product. Possible meanings: • D1 – Daytime • D2 – Evening • DD – Day • MA- Maximum load • MI – Minimum load • N1 – Night • N2 – Morning • NK – Nighttime, Saturday and Sunday • RG – Reactive power – generated • RV – Reactive power – consumed • SV – Holidays and rest days • VD – Medium load • VK – Single-rate
objectAddress	string	Y	Full address of the object.
contractType	string	Y	Contract type of the current object. Possible values: • SBTS - Household customer • SKMS - Commercial customer
contractModel	string	N	Contract model of the current object. Possible values: • BSS - General contract bills • 2S2S - Two contracts – Two bills
supplierType	string	Y	Current supplier type. Possible values: • VT - public supplier • GT - warranty supplier • NT - independent supplier
timeLimitedObjectValidTo	string (date)	N	Expiration date of the fixed-term object.

Attribute	Type	Mandatory	Description
tariffPlan	string	N	Current tariff plan of the object.
tariffPlanChangeDate	string (date)	N	The earliest date when the current object is eligible for a tariff plan change.
timeZone	string	N	Current time zone of the object. Possible meanings: • 1 - One • 2 - Two • VR - One with reactive • 4 - Four (Smart) • DR - Differentiated with reactive • N - Not established
ownershipDocumentNumber	string	N	Ownership document number.
supplyOwnership	string	N	Possible values: • MY – The object belongs to the interested party. • NOT_MY – The object does not belong to the interested party.
contractStart	string (date)	N	Contract start date with the interested party. • Returns only if: supplyOwnership = MY. • Otherwise: NULL is returned.
contractEnd	string (date)	N	Contract end date with the interested party. • Returns only if: supplyOwnership = MY. • Otherwise: NULL is returned.
accountingType	string	Y	Accounting type. Possible values: • NET_METERING – accumulates kwh • NET_BILLING – accumulates Eur • NET_METERING_NET_BILLING - accumulates kwh and Eur

Attribute	Type	Mandatory	Description
			• POWER_PLANT - sells kwh • CONSUMER - only consuming • ENERGY_SHARER – sharing kw
objectFutureSuppliers: []			
supplierCode	string	N	Future independent supplier code.
supplierName	string	N	Future independent supplier name.
contractStart	string (date)	N	Contract start date with the future independent supplier.
contractEnd	string (date)	N	Contract end date with the future independent supplier.
usedPowerPlants: []			
powerPlantObjectNumber	string	N	Unique number identifying the power plant object used for energy generation.
powerPlantType	string	N	Type of the power plant. Possible values: • A – Waste fuel • B – Biomass • H – Hydroelectric • K – Other • S – Solar • T – TEC • V – Wind • P – Storage device • I – Fossil • D – Biogas • R – Hybrid generation

Attribute	Type	Mandatory	Description
generatingObjectType	string	N	Type of generating consumer of the used power plant. Possible values: • G - Generating consumer • N - Distant generating consumer
generatingPower	number	N	Generation capacity of the assigned power plant.
powerPlantValidFrom	string (date)	Y	Generating consumer type valid from.
powerPlantValidTo	string (date)	N	Generating consumer type valid to.
accountingScheme	string	N	Generating consumer accounting scheme. Possible values: • NET_BILLING • NET_METERING
accountingSchemeValidFrom	string (date)	N	Generating consumer accounting scheme valid from.
accountingSchemeValidTo	string (date)	N	Generating consumer accounting scheme valid to.
accountingSchemeChangeDate	string (date)	N	Eligible date for accounting scheme change.
payoffMethod	string	N	Generating consumer payoff method. Possible values: • E – kWh – Recovered electricity • G - kW – Permitted capacity of the power plant • P - % - Payment percentage • S – kWh – PP recovered electricity
payoffMethodChangeDate	string (date)	N	The date from which the consumer is eligible to change payoff method.
objectPowers: []			

Attribute	Type	Mandatory	Description
powerType	string	N	Object power type. Possible values: • LOG - Permitted power consumption – the maximum allowed power the object is permitted to consume (in kW). • LGG - Permitted power generation – the maximum allowed power the object is permitted to generate (in kW). • IOG - Installed usable power – the actual installed capacity available for consumption (in kW). • IGG - Installed generation power – the actual installed capacity available for generation (in kW).
power	number	N	Object power
powerValidFrom	string (date)	N	Object power valid from.
powerValidTo	string (date)	N	Object power valid to.
generatingObjectGroup: {}			
generatingGroup	integer	N	The group identifier of the generating user.
generatingObjectPriorityGroup	integer	N	The priority of the generating user group object.
objectLatestSupplyState: {}			
state	string	N	Power supply state of the object. Possible values: • T – Supply • P – Disconnected on request • A – Disconnected under sanction • R – Limited by sanction
stateValidFrom	string (date)	N	Object state valid from.
stateValidTo	string (date)	N	Object state valid to.

Attribute	Type	Mandatory	Description
contact: {}			
mobPhoneNoNetwork	string	N	The mobile phone number associated with the object for network-related communication. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.
mobPhoneNo2Network	string	N	Extra mobile phone number for the network. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.
mobPhoneInvoice	string	N	Mobile phone number for invoice. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.
phoneNoNetwork	string	N	Phone number for network. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.
emailNetwork	string	N	Email address for the network. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.
emailNetwork2	string	N	Extra email address for the network. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.
emailInvoice	string	N	Email address for the invoice. • Data must be returned fully encrypted (***) if the contract type is SKMS. • Data must not be returned if the contract type is SBTS.

Attribute	Type	Mandatory	Description
consumptionAverage	number	N	Consumption average.
consumptionAverageCalculationDate	string (dateTime)	N	Date of consumption average calculation.
consumptionAverageCalculationMonthsCount	integer	N	Months count of consumption average calculation.
powerPlantObjectType	string	N	The object's power plant type. Possible meanings: • A – Waste fuel • B – Biomass • H – Hydroelectric • K – Other • S – Solar • T – TEC • V – Wind • P – Storage device • I – Fossil • D – Biogas • R – Hybrid generation