Data Export

Introduction

This article describes how to configure Wcc Lite to export data to external servers step by step.

Data export functionality is available since firmware version v1.5.0, of WCC Lite.

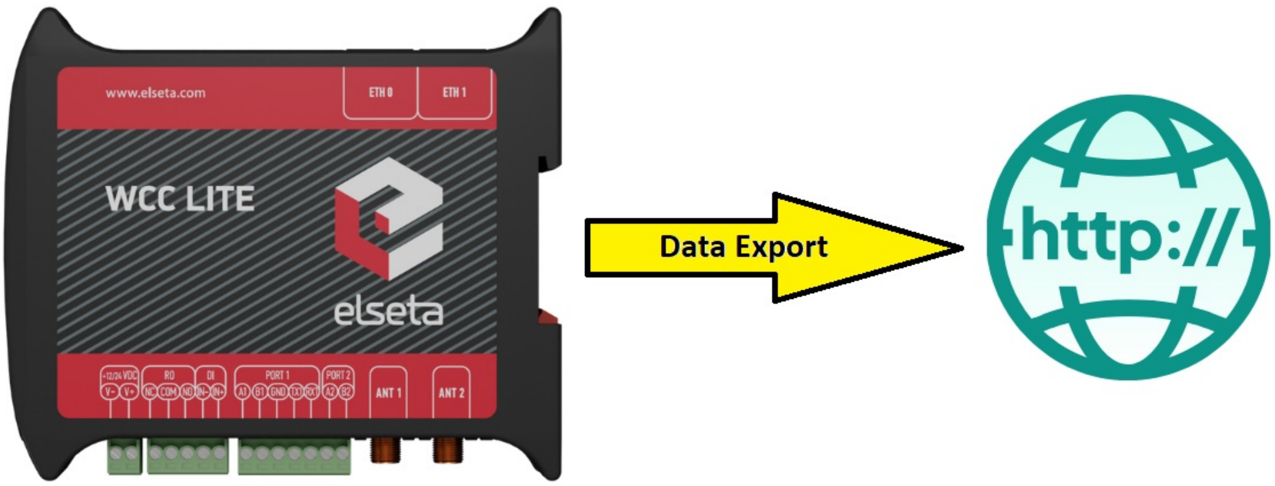


Fig. 1 Wcc Lite's gathered data export to external server

First step: Configuring the Devices sheet of excel configuration

To fully configure the "Data Export" device and it's signals, user can use the tables provided in the manual of Wcc Lite -> Data Export section or use the tables below in this article. Important to notice that some parameters to declare the Data Export device and it's signals are required and some of them are optional. In this article we will mainly use the required ones and a few of the optional parameters, to reach the desired outcome of signals exporting. Also, some parameters has default values, even if the parameters are not required, so please make sure that the default value is the same as needed, otherwise change it to the correct one from the available choices and describe it in excel configuration.

Data export (data-export) parameters for Devices tab table:

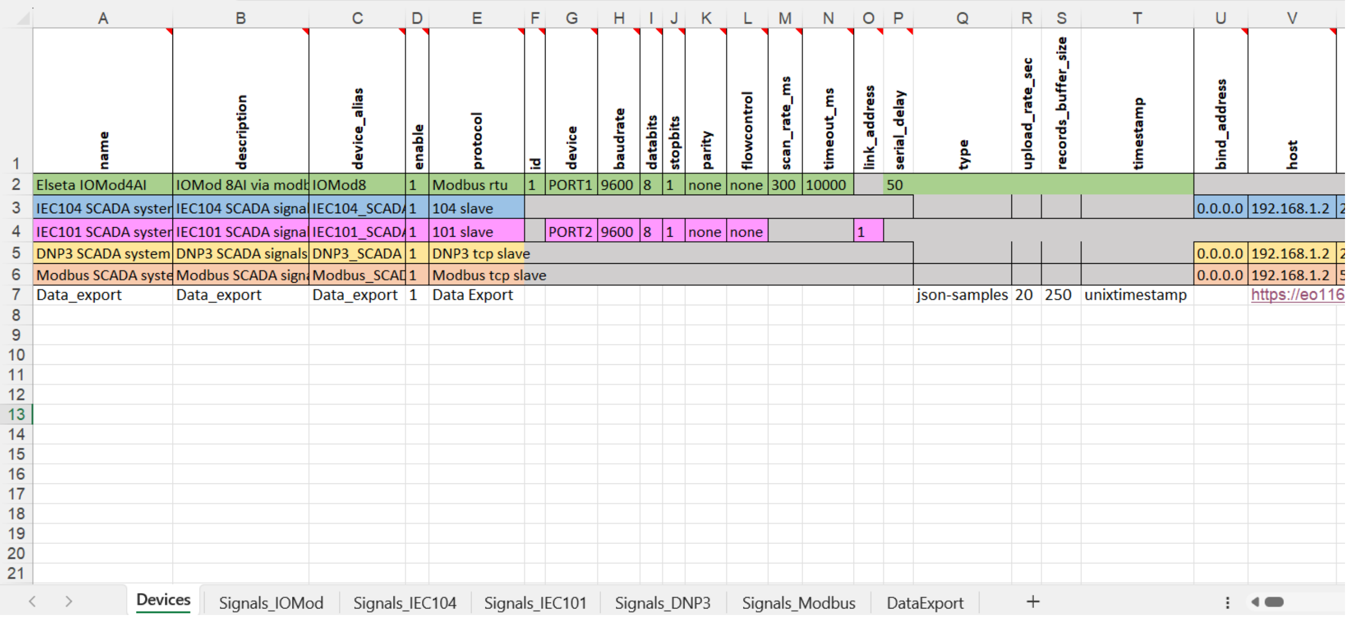
Parameter	**Type**	**Description**	**Required**	**Default value** (when not specified)	**Range**	
					Min	Max
name	string	User-friendly device name	Yes			
device_alias	string	Device alias to be used in configuration	Yes			
enable	boolean	Enabling/disabling of a device	No	1	0	1
protocol	string	Selection of protocol	Yes		Data Export	

timeout	integer	The time frame during which transmission to the remote server has to be completed (**in seconds**)	No	5		
type	string	Selection of file format	No	csv-simple	cvs-periodic csv-simple, json-simple, json-samples	
host	string	A URL of the remote server where files should be sent	Yes			
upload_rate_sec	integer	Frequency of generated file uploads (**in seconds)**	No	60		
records_buffer_size	integer	A maximum amount of data change entries to hold before initiating the logging mechanism	No	100		
logging_period_sec	integer	Describe how frequently the data buffer of records_buffer_size is saved to the file	No	10	1	3600
log_folder	string	A folder in the WCC Lite file system to save generated files (**”/var/cache/data-export”**)	No			
timestamp	string	Selection of time format	No	unixtimestamp	unixtimestamp, iso8601	
compress	string	Selection of file compression mechanism	No	none	none, gz, tar.gz	
compress_password	string	Enable the feature of file password protection	No		yes, true	
csv_field_separator	string	Columns separator in .csv file format	No	"," - (comma)	"," - (comma) ";" - (semicolon) "." - (dot) " " - whitespace) "|" - (pipe)	
csv_decimal_separator	string	Decimal separator in values	No	"." - (dot)	"." - (dot) "," - (comma)	

The same symbols cannot be selected for both csv_field_separator and csv_decimal_separator. In such case both of them will be set to default values ”.” and ”,” respectively.

The data generation rate may be going to be bigger than what the data buffer can hold (controlled by *records_buffer_size* and *logging_period_sec*). To make sure that no data loss occurs there’s an additional data logging call made in case the data buffer reaches a *records_buffer_size* value.

For this example configuration, the Wcc Lite's template configuration for IOMod 8AI Modbus RTU protocol is being used. Simply in Devices sheet add the information required for Data Export service. As it can be seen in **Fig. 2** to declare the Data Export device in Devices sheet a few parameters were used: ***name***, ***description***, ***device_alias***, ***enable***, ***protocol***, ***type***, ***upload_rate_sec***, ***record_buffer_size***, ***timestamp*** and ***host***.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	name	description	device_alias	enable	protocol	id	device	baudrate	databits	stopbits	parity	flowcontrol	scan_rate_ms	timeout_ms	link_address	serial_delay	type	upload_rate_sec	records_buffer_size	timestamp	bind_address	host	
2	Elseta IOMod4AI	IOMod 8AI via modb	IOMod8	1	Modbus rtu	1	PORT1	9600	8	1	none	none	300	10000		50							
3	IEC104 SCADA syster	IEC104 SCADA signal	IEC104_SCADA	1	104 slave																0.0.0.0	192.168.1.2	2
4	IEC101 SCADA syster	IEC101 SCADA signal	IEC101_SCADA	1	101 slave		PORT2	9600	8	1	none	none			1								
5	DNP3 SCADA system	DNP3 SCADA signals	DNP3_SCADA	1	DNP3 tcp slave																0.0.0.0	192.168.1.2	2
6	Modbus SCADA syste	Modbus SCADA signa	Modbus_SCAD	1	Modbus tcp slave																0.0.0.0	192.168.1.2	5
7	Data_export	Data_export	Data_export	1	Data Export												json-samples	20	250	unixtimestamp		https://eo116	
8																							
9																							
10																							
11																							
12																							
13																							
14																							
15																							
16																							
17																							
18																							
19																							
20																							
21																							

Fig. 2 Describing the Data Export service in Devices sheet of excel configuration

Host URL format rules

The parameter host is highly configurable and might contain a considerable amount of information:

- *Protocol* - FTP or HTTP (encrypted and encrypted);

- *URL address* - both resolved and non-resolved;

- *Authentication* - pair of users and/or passwords;

- *Port* - useful when non-standard value is used;

- *Endpoint* - a place in the server to which a call is made

The format for the host parameter can be summarized as:

```
[http(s)/ftp(s)]://[[user]:[password]@] [ URL ] [:port]/[endpoin
]
```

Options are printed in square brackets. A protocol has to be selected, otherwise HTTP will be used as a default. The user and password pair is optional, but if the user: password pair is used, it should proceed with the @ sign.

HTTP and FTP use default or user-assigned ports. By default HTTP uses port 80, while HTTPS uses port 443, FTP sends data over port 21, FTPS - over port 990. Make sure that these ports are open in the firewall on both the server and client side, otherwise, data will not be sent successfully.

Finally, a POST request (for HTTP) or upload (for FTP) can be made to a specific place (endpoint). This endpoint should be described after a URL and port (if used).

Second step: Configuring the DataExport sheet of excel configuration

Signals to be sent are configured differently than signals for most other protocols As data export service only transmits signals and does no data processing, usual signal logic is not used for them. That means that:

- Signals for data export service are not seen in the*Imported Signals* tab;
- Signals for data export service are configured in a different Excel sheet called DataExport

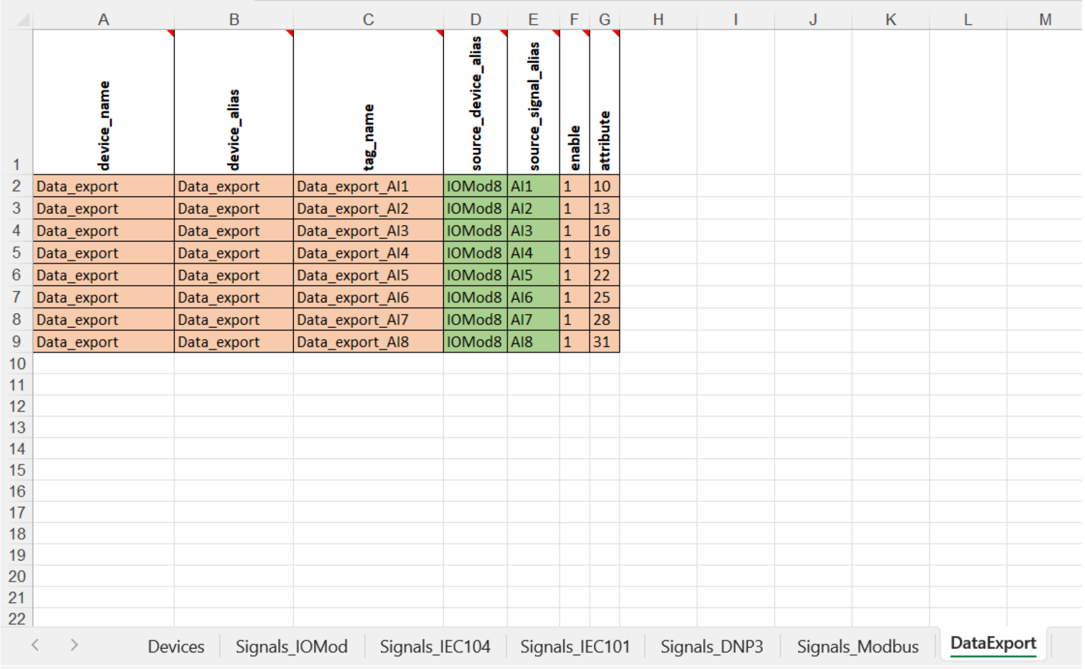
The parameters to be filled in the DataExport sheet are shown in the table below.

Data export (data-export) parameters for the DataExport tab

Parameter	Type	Description	Required	**Default value** (when not specified)	Range	
					Min	Max

device_alias	string	Device alias to be used in configuration Yes	Yes			
device_name	string	User-friendly device name as in the Device sheet	Yes			
tag_name	string	User-friendly signal name	Yes			
source_device_alias	string	device_alias of a source device	Yes			
source_signal_alias	string	source_alias of a source signal	Yes			
enable	boolean	Enabling/disabling of a measurement to be transmitted and logged	No	1	0	1
attribute	string	An additional attribute to be attached to a signal	No			

To configure Wcc Lite to export data to an external server user must declare the signals for every measurement that needs to be exported, as well as describe the device (Data Export) in Devices sheet. This is done simply by linking signals from other protocols to the signals declared for exporting data. In **Fig. 3** it is shown how measurements from IOMod 8AI (Signals_IOMod sheet ***device_alias*** and ***signal_alias***) are linked to the signals (DataExport sheet ***source_device_alias*** and ***source_signal_alias***) that will be exported to the server, which is declared as "Host" parameter in Devices sheet of excel configuration.



device_name	device_alias	tag_name	source_device_alias	source_signal_alias	enable	attribute
Data_export	Data_export	Data_export_AI1	IOMod8	AI1	1	10
Data_export	Data_export	Data_export_AI2	IOMod8	AI2	1	13
Data_export	Data_export	Data_export_AI3	IOMod8	AI3	1	16
Data_export	Data_export	Data_export_AI4	IOMod8	AI4	1	19
Data_export	Data_export	Data_export_AI5	IOMod8	AI5	1	22
Data_export	Data_export	Data_export_AI6	IOMod8	AI6	1	25
Data_export	Data_export	Data_export_AI7	IOMod8	AI7	1	28
Data_export	Data_export	Data_export_AI8	IOMod8	AI8	1	31

Fig. 3 Linking of the measurements from IOMod 8AI to Data Export signals

Data representation in HTTP(S) server

First thing to mention, that if user wants to export the data to an external server, the Wcc Lite should have access to the internet. This is done simply by connecting Wcc Lite's ETH1 port to the Ethernet port of the internet router.

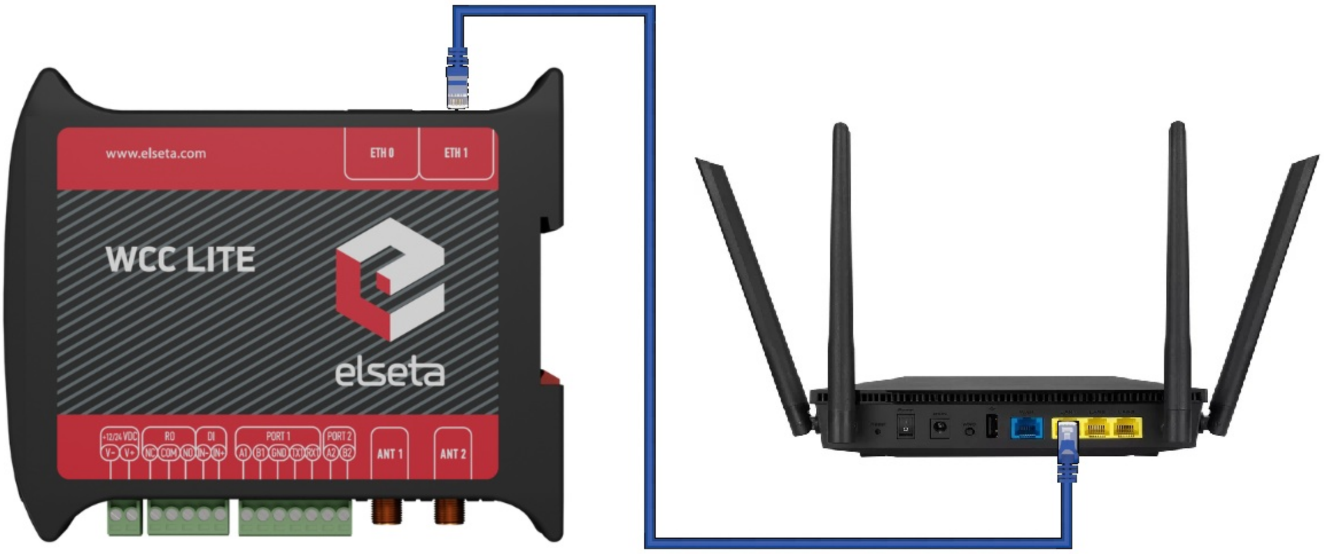
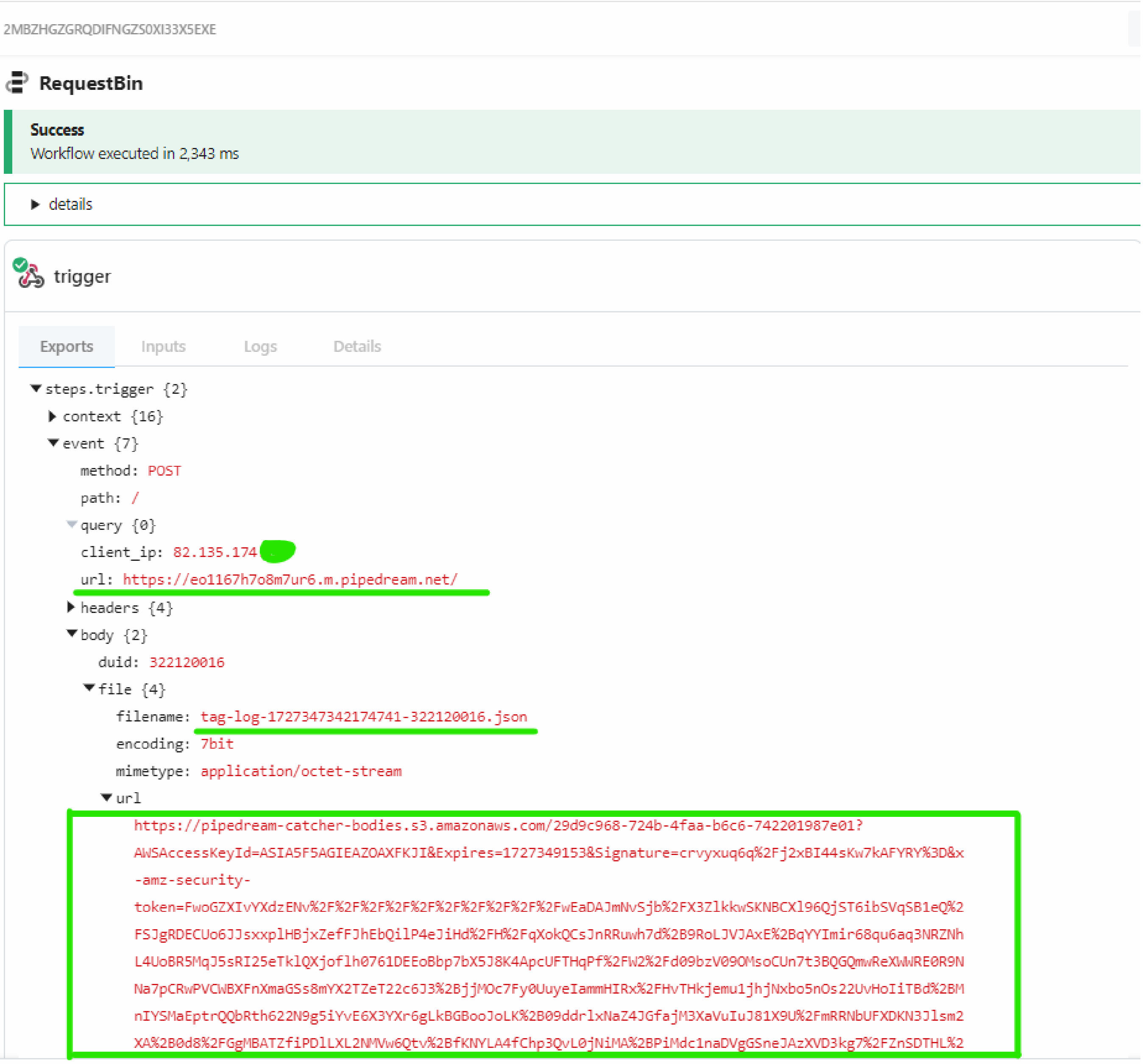


Fig. 4 Wcc Lite's ETH1 port connected to the internet router for internet access

For the representation of data in HTTP(S) or FTP(S) servers, user can use one of the available servers that can be found online. In this specific example Pipedream.com online HTTP server was used. When registered to the site, generate the URL for the server. This URL can be seen in example configuration Devices tab "host" parameter for Data Export device (*https://eo1167h7o8m7ur6.m.pipedream.net).* When the communication between Wcc Lite and IOMod is normal, and Wcc Lite is gathering measurements through Modbus RTU protocol, these measurement, as described before, are linked to Data Export signals. These signal values then are exported to the predefined HTTP(S) server in a defined file format.

On the Pipedream.com online server a POST request (for HTTP) is being done and the data has been exported to it. As it can be seen in the **Fig. 5**, first URL is the same as the one declared in "Host" parameter for the Data Export service. Then the File and it's name, where the gathered data is being put and the URL on the bottom to download this file. How these measurements are represented in json_samples format can be seen in **Fig. 6** and in the *tag-log-1727347342174741-322120016.json* file on the bottom of this article. Also to see other file formats for exported data please check Data Export article in Wcc Lite manual.



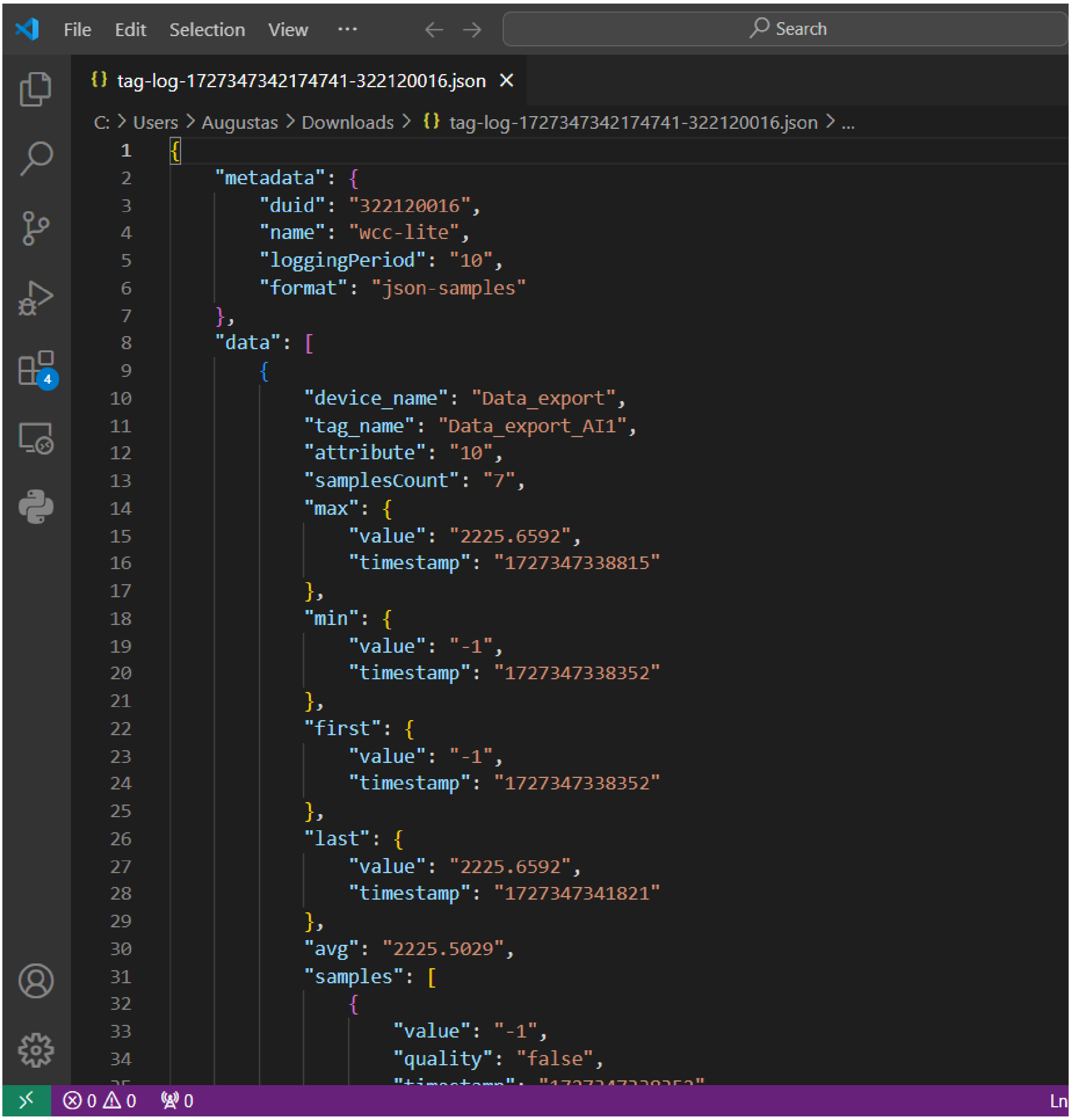
2MBZHGZGRQDIFNGZS0XI33X5EXE

RequestBin

Success
Workflow executed in 2,343 ms

```
▼steps.trigger {2}
  ▶context {16}
  ▼event {7}
     method: POST
     path: /
    ▼query {0}
     client_ip: 82.135.174[redacted]
     url: https://eo1167h7o8m7ur6.m.pipedream.net/
    ▶headers {4}
    ▼body {2}
       duid: 322120016
      ▼file {4}
         filename: tag-log-1727347342174741-322120016.json
         encoding: 7bit
         mimetype: application/octet-stream
        ▼url
           https://pipedream-catcher-bodies.s3.amazonaws.com/29d9c968-724b-4faa-b6c6-742201987e01?
           AWSAccessKeyId=ASIA5F5AGIEAZOAXFKJI&Expires=1727349153&Signature=crvyxuq6q%2Fj2xBI44sKw7kAFYRY%3D&x
           -amz-security-
           token=FwoGZXIvYXdzENv%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDAJmNvSjb%2FX3ZlkkwSKNBCXl96QjST6ibSVqSB1eQ%2
           FSJgRDECUo6JJsxxplHBjxZefFJhEbQilP4eJiHd%2FH%2FqXokQCsJnRRuwh7d%2B9RoLJVJAxE%2BqYYImir68qu6aq3NRZNh
           L4UoBR5MqJ5sRI25eTklQXjoflh0761DEEoBbp7bX5J8K4ApcUFTHqPf%2FW2%2Fd09bzV09OMsoCUn7t3BQGQmwReXWWRE0R9N
           Na7pCRwPVCWBXFnXmaGSs8mYX2TZeT22c6J3%2BjjMOc7Fy0UuyeIammHIRx%2FHvTHkjemu1jhjNxbo5nOs22UvHoIiTBd%2BM
           nIYSMaEptrQQbRth622N9g5iYvE6X3YXr6gLkBGBooJoLK%2B09ddrlxNaZ4JGfajM3XaVuIuJ81X9U%2FmRRNbUFXDKN3Jlsm2
           XA%2B0d8%2FGgMBATZfiPDlLXL2NMVw6Qtv%2BfKNYLA4fChp3QvL0jNiMA%2BPiMdc1naDVgGSneJAzXVD3kg7%2FZnSDTHL%2
```

Fig. 5 Information in Pipedream.com HTTP server about the data packet exported from Wcc Lite



File
Edit
Selection
View
···
Search
tag-log-1727347342174741-322120016.json
C: > Users > Augustas > Downloads > tag-log-1727347342174741-322120016.json > ...
4
0
0
0
Ln

```json
{
    "metadata": {
        "duid": "322120016",
        "name": "wcc-lite",
        "loggingPeriod": "10",
        "format": "json-samples"
    },
    "data": [
        {
            "device_name": "Data_export",
            "tag_name": "Data_export_AI1",
            "attribute": "10",
            "samplesCount": "7",
            "max": {
                "value": "2225.6592",
                "timestamp": "1727347338815"
            },
            "min": {
                "value": "-1",
                "timestamp": "1727347338352"
            },
            "first": {
                "value": "-1",
                "timestamp": "1727347338352"
            },
            "last": {
                "value": "2225.6592",
                "timestamp": "1727347341821"
            },
            "avg": "2225.5029",
            "samples": [
                {
                    "value": "-1",
                    "quality": "false",
                    "timestamp": "1727347338352"
```

Fig. 6 Measurements exported in json-samples format

Files used in this article:

1. IOMod8AI_data_export.xlsx

2. tag-log-1727347342174741-322120016.json

⟲Revision #5
★Created 26 September 2024 11:27:50 by Augustas
✎Updated 30 September 2024 11:32:19 by Augustas