DCF

Data Collection Framework

User Manual

SHOP4CF

Tuan Vu (TAU)

Contents

INTRODUCTION

The purpose of this manual is to introduce the architecture of DCF as well as steps to install DCF and test the functionalities, the manual also guides the users on how to navigate the web GUI and correctly define the input accordingly to the data formats.

Requirement:

- PC
- Docker
- DCF image
- Web browser (Edge, Firefox, etc)

For testing (optional):

- Either one of: Postman or programming IDE for performing CRUD operations
- Scripts for importing data from csv files to Mongodb for testing

DCF ARCHITECTURE

DCF role is to collect data from shopfloor through data adapter for example MQTT, OPC-UA, Fiware/orion or legacy system and data stored in legacy systems and database for example mongodb, after that the data is transmitted through Fiware/orion.

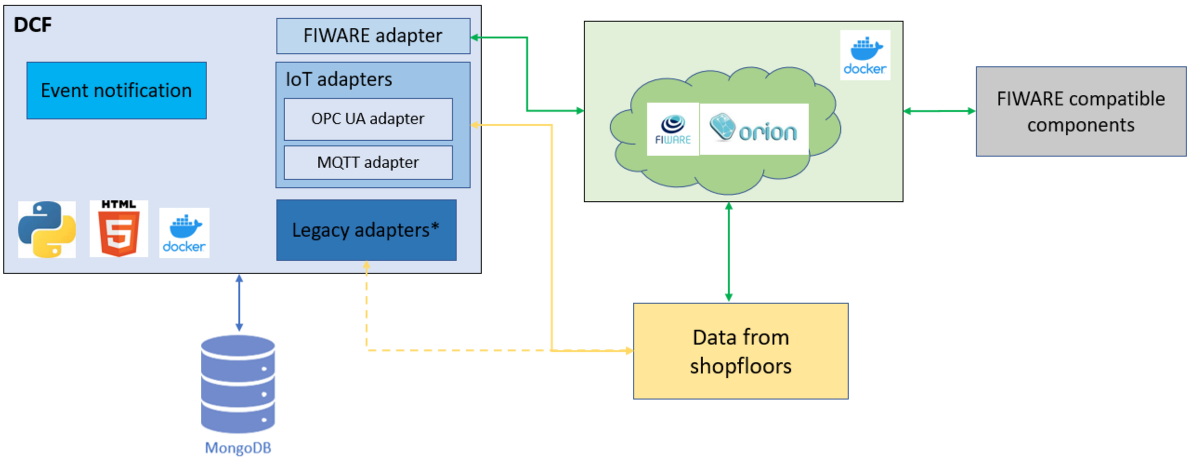


Figure 1: DCF architecture

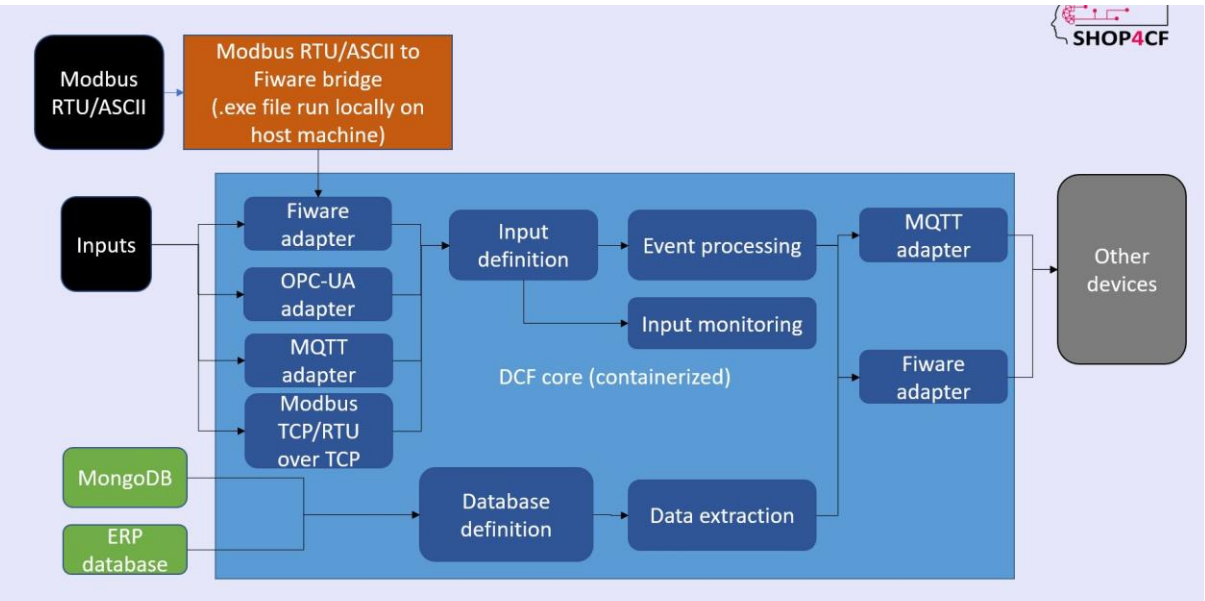


Figure 2: DCF supported communication protocols

To interact with other components, FIWARE is needed, although, DCF can be configurated to communicated by using other brokers/adapters for example OPC-UA and MQTT as well.

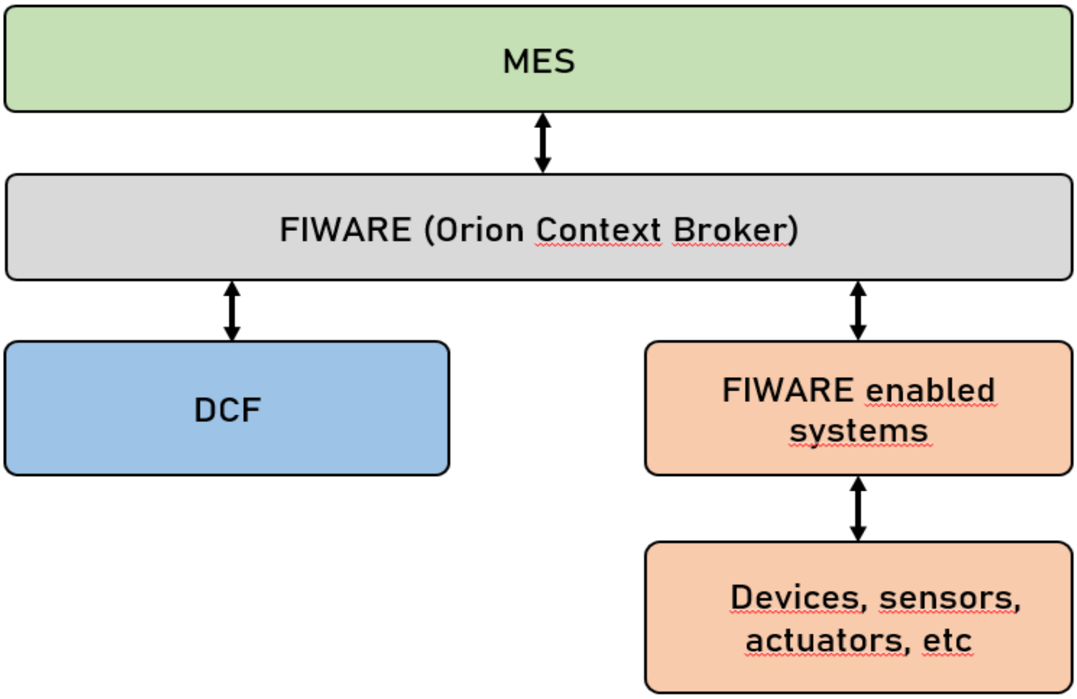
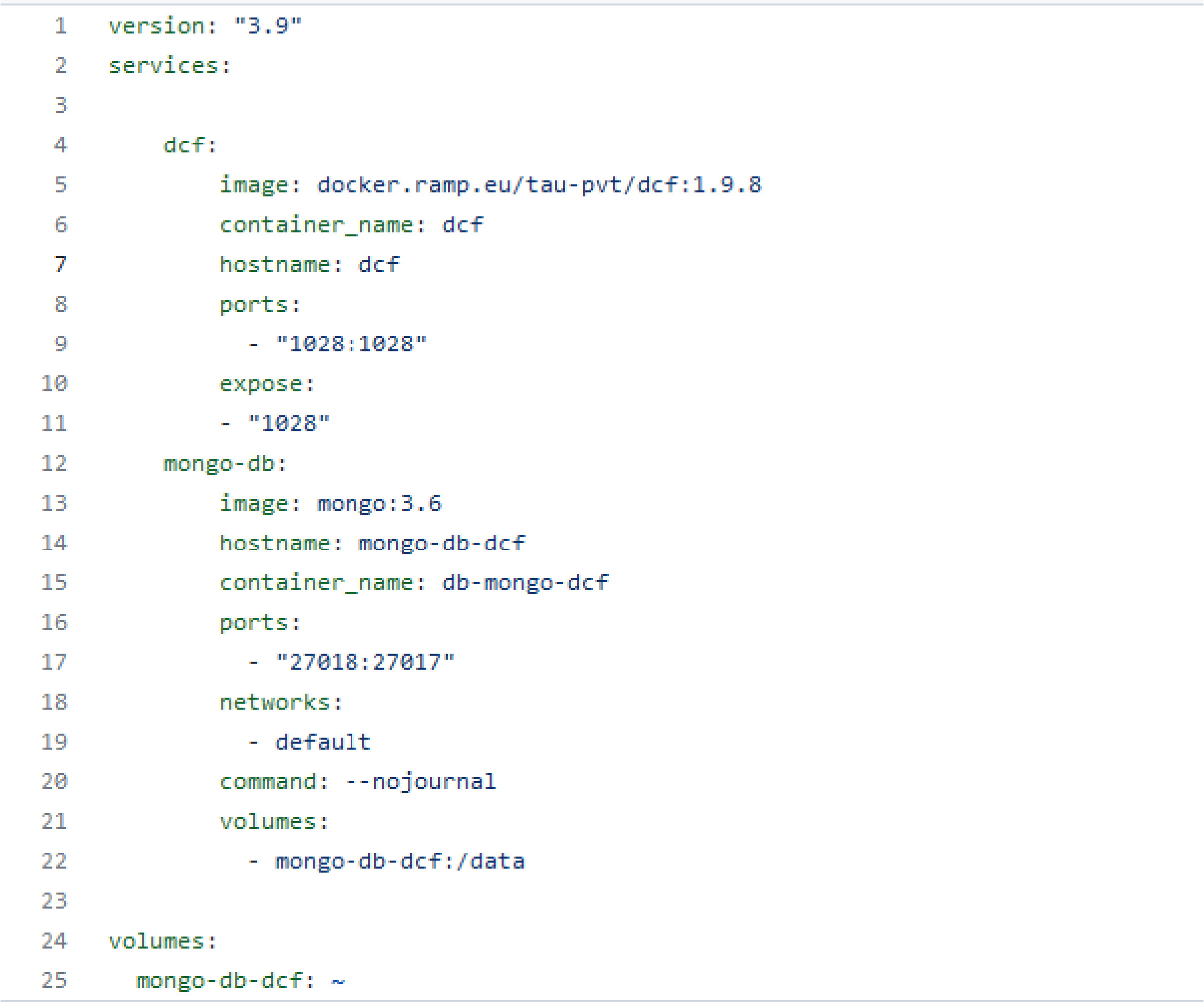

MES
FIWARE (Orion Context Broker)
DCF
FIWARE enabled systems
Devices, sensors, actuators, etc

Figure 3: DCF communication with other components

DOWNLOADS ADDITIONAL FILES AND INSTALLATION

Beside DCF image, only docker-compose file is needed to compose the image into container. To get the docker-compose.yml file, download the file from https://github.com/TAU-FASTLab/DCF. Version of DCF can be changed to compose the suitable version. The latest version is 1.9.8. The sample docker-compose file can be seen below, the version of component can be changed by changing the tag number of image. DCF is also supported by its own Mongo-DB module, which stores input data, the module is also defined in the docker-compose file.

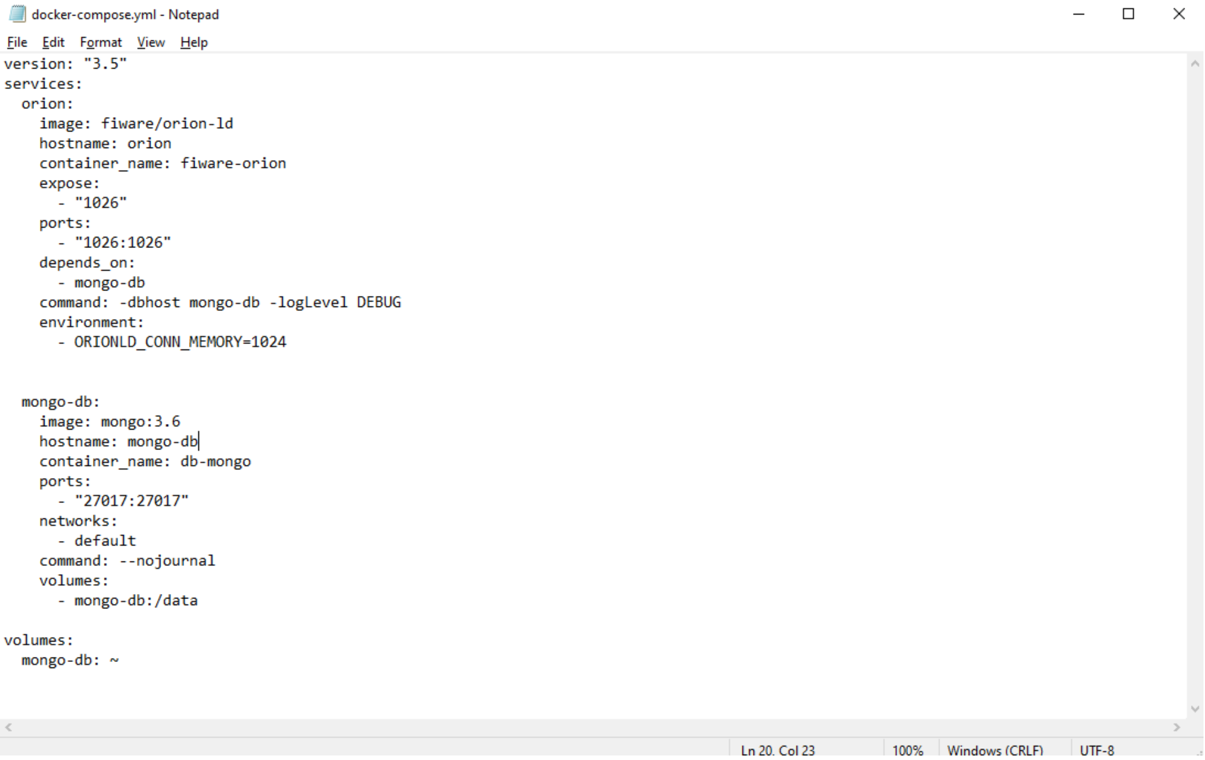


```yaml
version: "3.9"
services:

    dcf:
        image: docker.ramp.eu/tau-pvt/dcf:1.9.8
        container_name: dcf
        hostname: dcf
        ports:
          - "1028:1028"
        expose:
        - "1028"
    mongo-db:
        image: mongo:3.6
        hostname: mongo-db-dcf
        container_name: db-mongo-dcf
        ports:
          - "27018:27017"
        networks:
          - default
        command: --nojournal
        volumes:
          - mongo-db-dcf:/data

volumes:
  mongo-db-dcf: ~
```

If users use dockerized fiware and MongoDB server or other applications, when referring to these servers for data retrieving and publishing, the host name will be “host.docker.internal” instead of “localhost”, the ports will be the same ports of these applications.

SETTING UP DOCKERIZED VERSION OF FIWARE/ORION-LD BROKER EXAMPLE

To setup Fiware/Orion-LD broker running on docker, the following docker-compose file is tested and recommended:



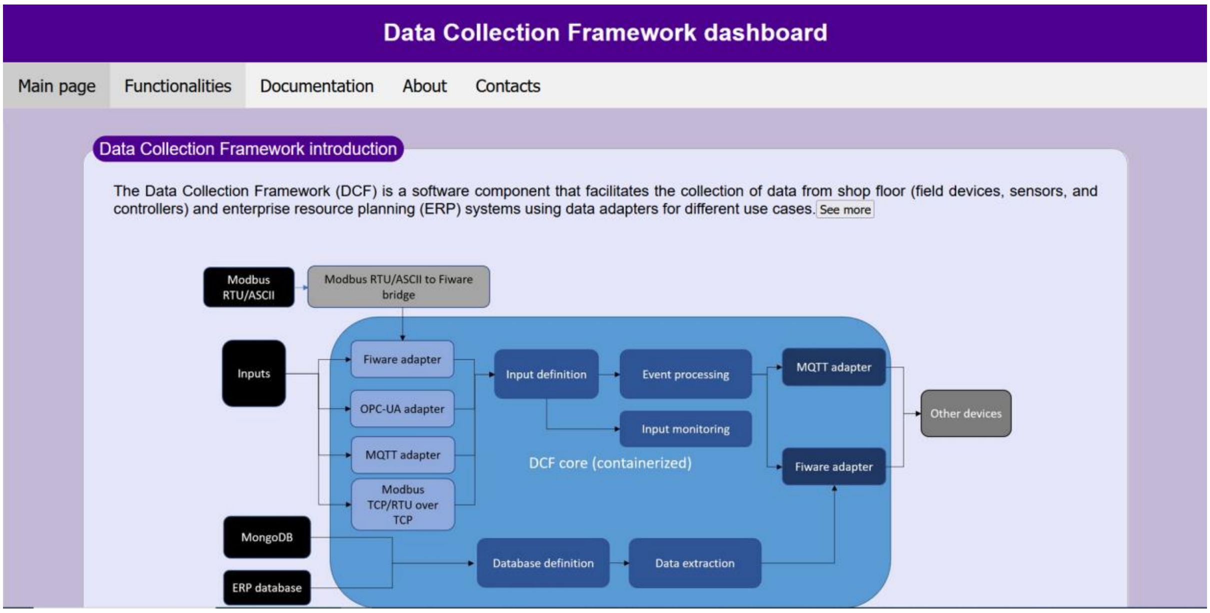
```
version: "3.5"
services:
  orion:
    image: fiware/orion-ld
    hostname: orion
    container_name: fiware-orion
    expose:
      - "1026"
    ports:
      - "1026:1026"
    depends_on:
      - mongo-db
    command: -dbhost mongo-db -logLevel DEBUG
    environment:
      - ORIONLD_CONN_MEMORY=1024


  mongo-db:
    image: mongo:3.6
    hostname: mongo-db
    container_name: db-mongo
    ports:
      - "27017:27017"
    networks:
      - default
    command: --nojournal
    volumes:
      - mongo-db:/data

volumes:
  mongo-db: ~
```

Environment parameters are entirely optional. Additional info can be found on docker hub page. https://hub.docker.com/r/fiware/orion-ld/

USER INTERFACE

DCF also provides a GUI where users can monitor events and define new operations.

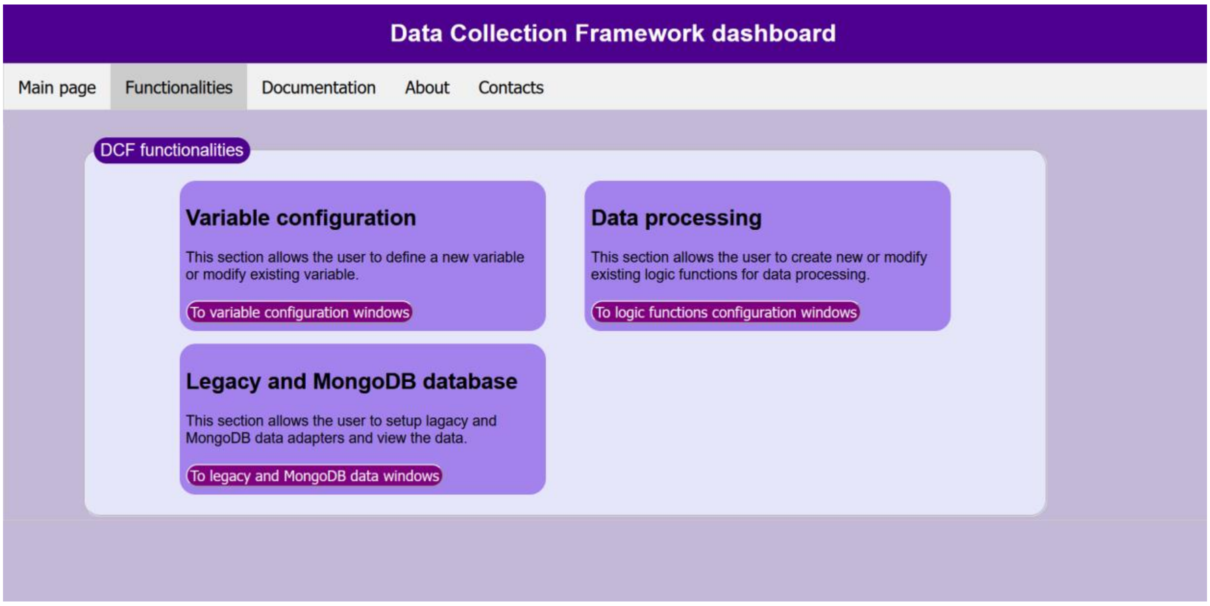
The front page is divided into section which can be accessed by selecting navigation tabs.



Data Collection Framework dashboard
Main page
Functionalities
Documentation
About
Contacts
Data Collection Framework introduction

The Data Collection Framework (DCF) is a software component that facilitates the collection of data from shop floor (field devices, sensors, and controllers) and enterprise resource planning (ERP) systems using data adapters for different use cases.

The Functionalities tab open list of functionalities supported by DCF.



Data Collection Framework dashboard

Main page
Functionalities
Documentation
About
Contacts
DCF functionalities

Variable configuration

This section allows the user to define a new variable or modify existing variable.

To variable configuration windows

Data processing

This section allows the user to create new or modify existing logic functions for data processing.

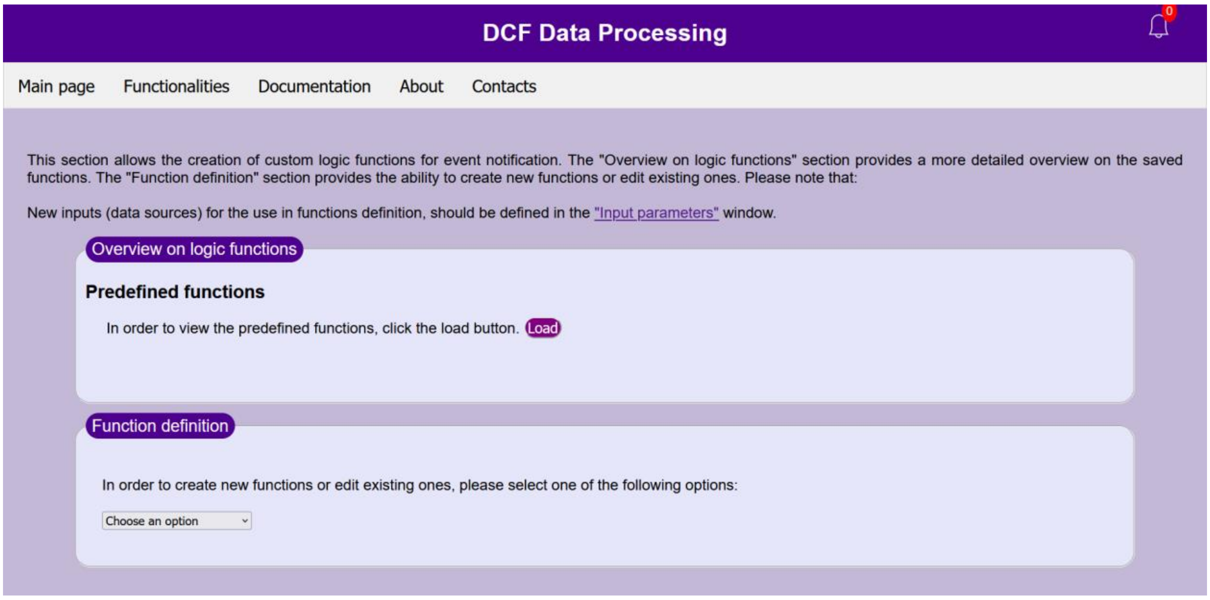
To logic functions configuration windows

Legacy and MongoDB database

This section allows the user to setup lagacy and MongoDB data adapters and view the data.

Data processing interface

This interface allows users to define new logic operations or modify existing operations as well as to receive notifications produced by the data processing function.



DCF Data Processing

This section allows the creation of custom logic functions for event notification. The "Overview on logic functions" section provides a more detailed overview on the saved functions. The "Function definition" section provides the ability to create new functions or edit existing ones. Please note that:

New inputs (data sources) for the use in functions definition, should be defined in the "Input parameters" window.

Overview on logic functions

Predefined functions

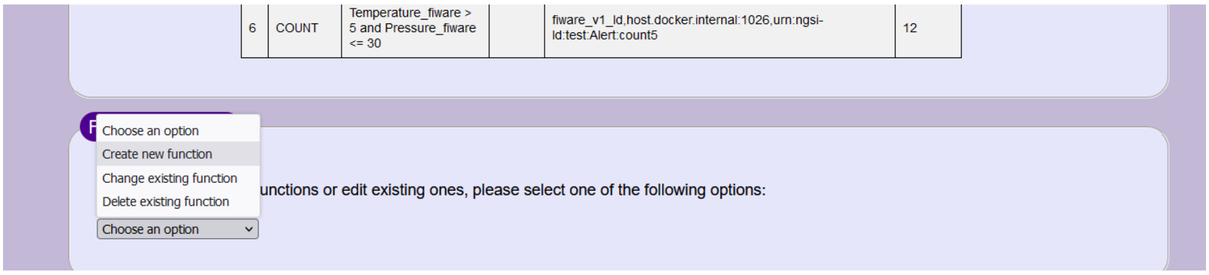
In order to view the predefined functions, click the load button. Load

Function definition

In order to create new functions or edit existing ones, please select one of the following options:

Choose an option

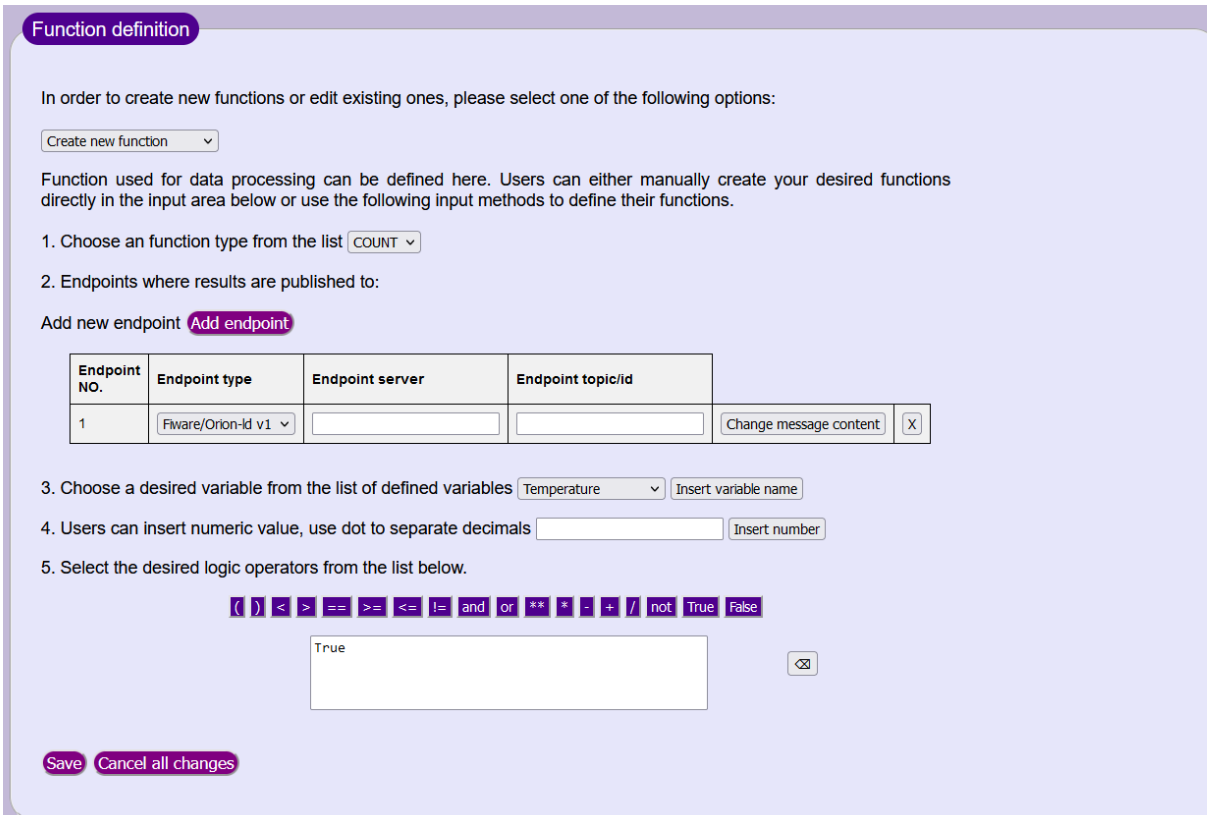
To create/modify/delete operation, select the dropdown menu and select one of the functionalities



6	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count5	12

unctions or edit existing ones, please select one of the following options:

Creating new operation



Function definition

In order to create new functions or edit existing ones, please select one of the following options:

Create new function

Function used for data processing can be defined here. Users can either manually create your desired functions directly in the input area below or use the following input methods to define their functions.

1. Choose an function type from the list COUNT

2. Endpoints where results are published to:

Add new endpoint Add endpoint

Endpoint NO.	Endpoint type	Endpoint server	Endpoint topic/id		
1	Fiware/Orion-ld v1			Change message content	X

3. Choose a desired variable from the list of defined variables Temperature Insert variable name

4. Users can insert numeric value, use dot to separate decimals Insert number

5. Select the desired logic operators from the list below.

```
True
```

In this section, users can choose which type of operation, ALARM type generate an output only when the counter of number of times the logic condition has been triggered exceeds or reaches the Alarm count limit parameter, while COUNT type constantly presents how many times the logic condition has been triggered.

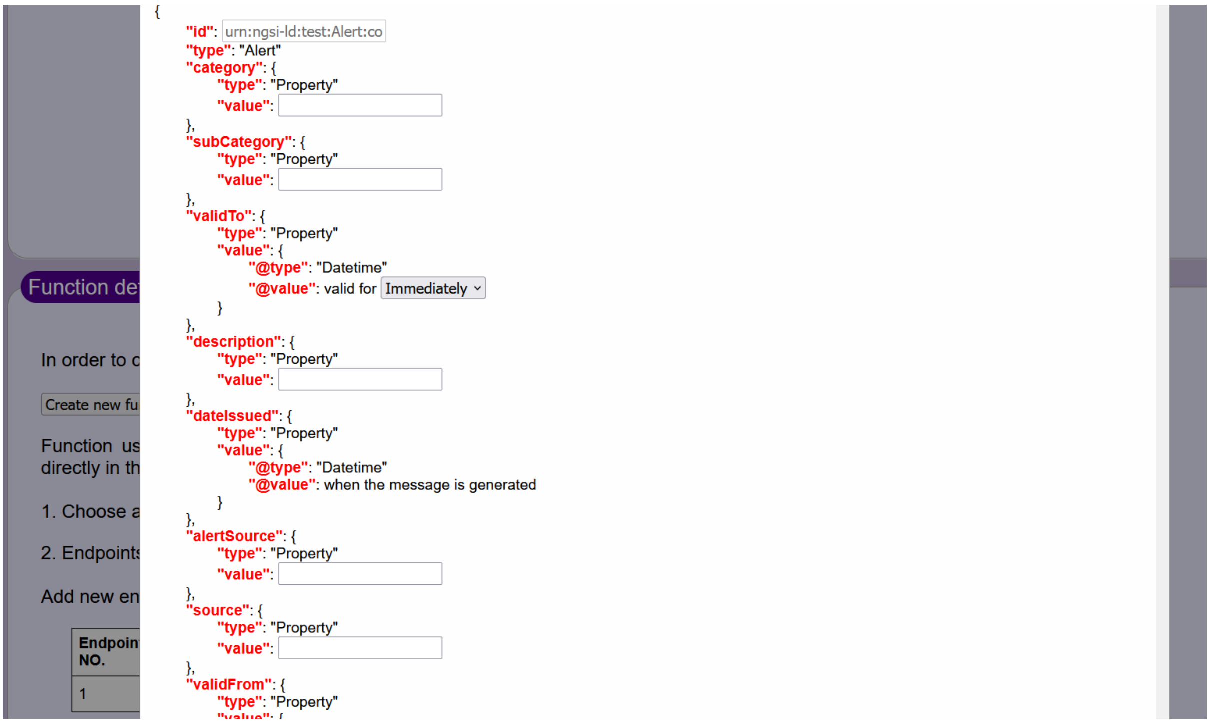
Section 2 is used to define the endpoints where processed results are published to

Section 5 is the place where logic arguments are created, the default condition for new operation is always “True”.

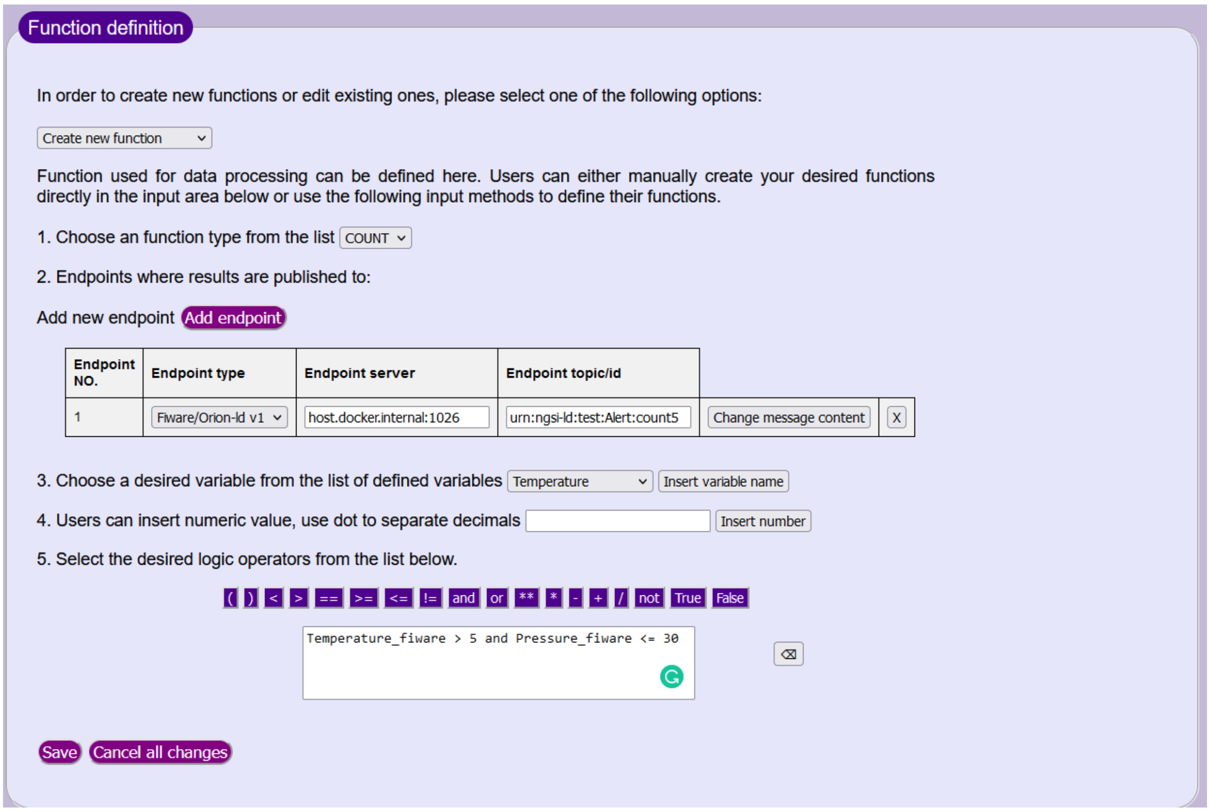
There is list of tools are created to assist users to create the logic condition, the list of defined variables dropdown menu contains all defined variables by users. After users has chosen a variable or given a numeric input (with “.” as decimal delimiter and no thousand separators), “insert variable”/ “insert number” need to be clicked respectively to insert the inputs. Users can also choose logic operators from the logic operator list. It is highly recommended to use these features to ensure the data format is followed so the program can parse the data. To finalize, the “confirm logic change” button is also needed to be chosen.

For example, the logic condition “Temperature_fiware > 5 and Pressure_fiware <= 30” is defined.

To publish the data to MQTT or Fiware/orion-ld, users need to use section 2 to define the endpoints, topic/id is recommended to be unique, endpoint type is one of “Fiware/Orion-ld v1” or “mqtt” or “Fiware v2”. In this case, the result needs to be publish to Fiware/Orion-ld v1 with the entity id “urn:ngsi-ld:test:Alert:count5”, on my containerized fiware/orion-ld server so the server should be host.docker.internal:1026. The content of the entities follows SHOP4CF data model but the values of content fields can be modified. If the fields are empty, default values will be used instead.



```
{
    "id": urn:ngsi-ld:test:Alert:co
    "type": "Alert"
    "category": {
        "type": "Property"
        "value": 
    },
    "subCategory": {
        "type": "Property"
        "value": 
    },
    "validTo": {
        "type": "Property"
        "value": {
            "@type": "Datetime"
            "@value": valid for Immediately ⌄
        }
    },
    "description": {
        "type": "Property"
        "value": 
    },
    "dateIssued": {
        "type": "Property"
        "value": {
            "@type": "Datetime"
            "@value": when the message is generated
        }
    },
    "alertSource": {
        "type": "Property"
        "value": 
    },
    "source": {
        "type": "Property"
        "value": 
    },
    "validFrom": {
        "type": "Property"
        "value": {
```



Function definition

In order to create new functions or edit existing ones, please select one of the following options:

Create new function ⌄

Function used for data processing can be defined here. Users can either manually create your desired functions directly in the input area below or use the following input methods to define their functions.

1. Choose an function type from the list COUNT ⌄

2. Endpoints where results are published to:

Add new endpoint Add endpoint

Endpoint NO.	Endpoint type	Endpoint server	Endpoint topic/id		
1	Fiware/Orion-ld v1 ⌄	host.docker.internal:1026	urn:ngsi-ld:test:Alert:count5	Change message content	X

3. Choose a desired variable from the list of defined variables Temperature ⌄ Insert variable name

4. Users can insert numeric value, use dot to separate decimals Insert number

5. Select the desired logic operators from the list below.

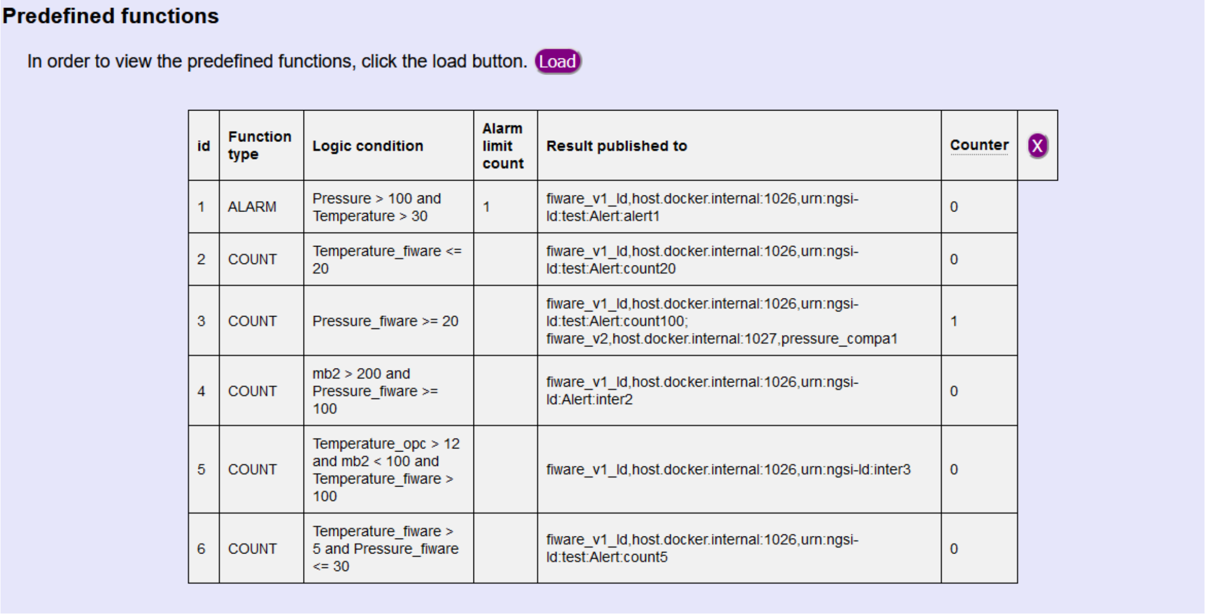
( ) < > == >= <= != and or ** * - + / not True False

```
Temperature_fiware > 5 and Pressure_fiware <= 30
```

⌫

Save Cancel all changes

Push “Save” button to update the data.

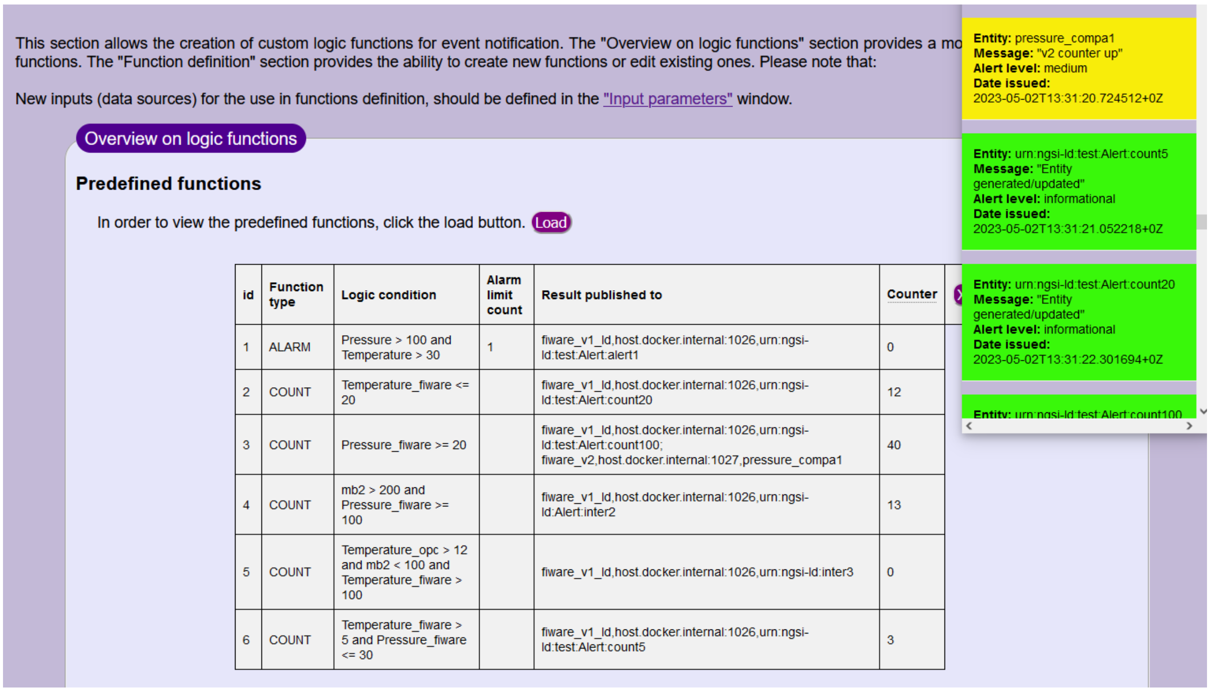


Predefined functions

In order to view the predefined functions, click the load button. Load

id	Function type	Logic condition	Alarm limit count	Result published to	Counter	X
1	ALARM	Pressure > 100 and Temperature > 30	1	fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:alert1	0	
2	COUNT	Temperature_fiware <= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count20	0	
3	COUNT	Pressure_fiware >= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count100; fiware_v2,host.docker.internal:1027,pressure_compa1	1	
4	COUNT	mb2 > 200 and Pressure_fiware >= 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:Alert:inter2	0	
5	COUNT	Temperature_opc > 12 and mb2 < 100 and Temperature_fiware > 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:inter3	0	
6	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count5	0	

To check whether the operation is operational, head to either the monitor dashboard or entity of the Alert in fiware/orion server using internet browsers or postman.



This section allows the creation of custom logic functions for event notification. The "Overview on logic functions" section provides a mo functions. The "Function definition" section provides the ability to create new functions or edit existing ones. Please note that:

New inputs (data sources) for the use in functions definition, should be defined in the "Input parameters" window.

Overview on logic functions

Predefined functions

In order to view the predefined functions, click the load button. Load

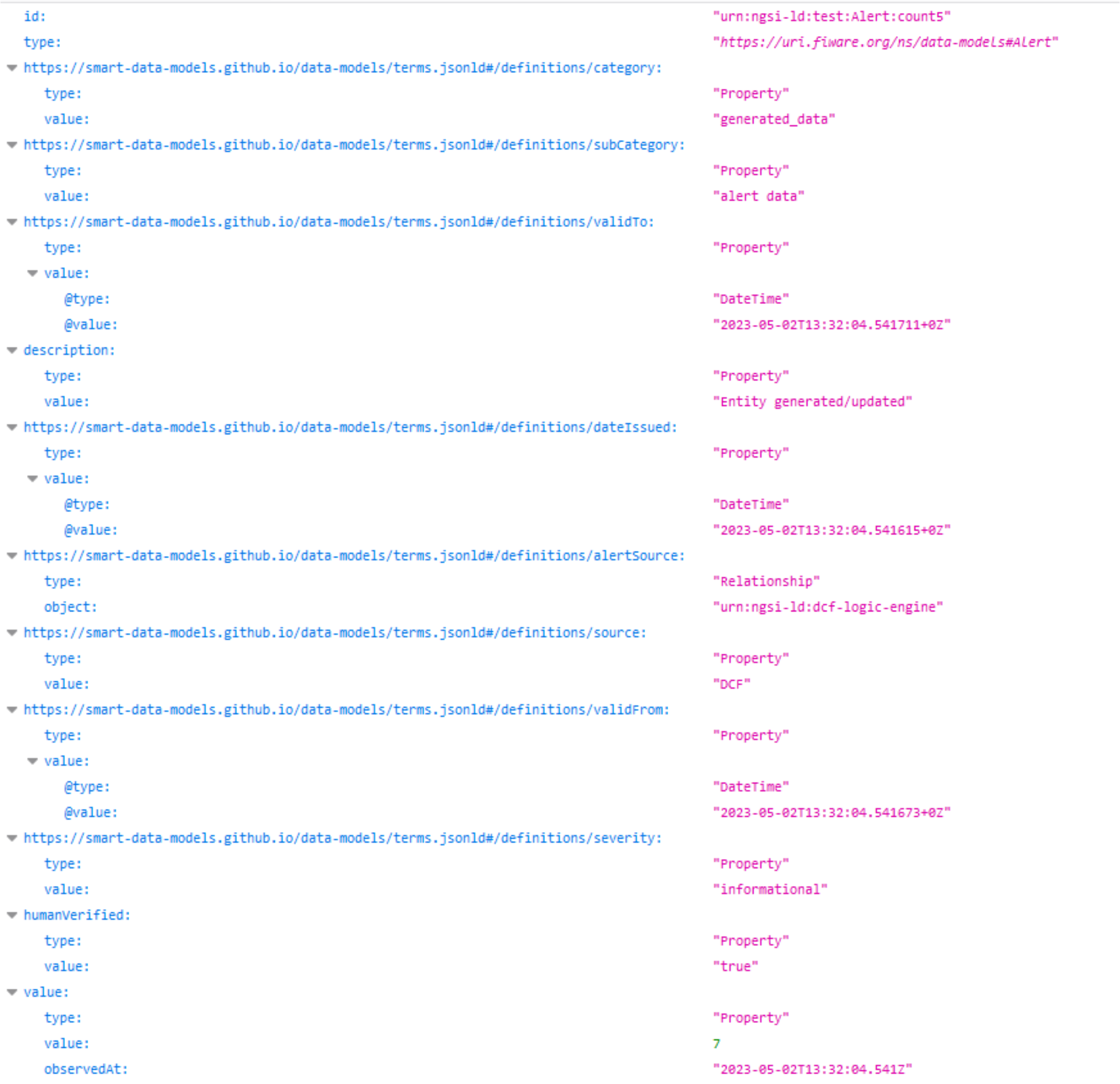
id	Function type	Logic condition	Alarm limit count	Result published to	Counter
1	ALARM	Pressure > 100 and Temperature > 30	1	fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:alert1	0
2	COUNT	Temperature_fiware <= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count20	12
3	COUNT	Pressure_fiware >= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count100; fiware_v2,host.docker.internal:1027,pressure_compa1	40
4	COUNT	mb2 > 200 and Pressure_fiware >= 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:Alert:inter2	13
5	COUNT	Temperature_opc > 12 and mb2 < 100 and Temperature_fiware > 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:inter3	0
6	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count5	3

Entity: pressure_compa1
Message: "v2 counter up"
Alert level: medium
Date issued: 2023-05-02T13:31:20.724512+0Z

Entity: urn:ngsi-ld:test:Alert:count5
Message: "Entity generated/updated"
Alert level: informational
Date issued: 2023-05-02T13:31:21.052218+0Z

Entity: urn:ngsi-ld:test:Alert:count20
Message: "Entity generated/updated"
Alert level: informational
Date issued: 2023-05-02T13:31:22.301694+0Z

Entity: urn:ngsi-ld:test:Alert:count100

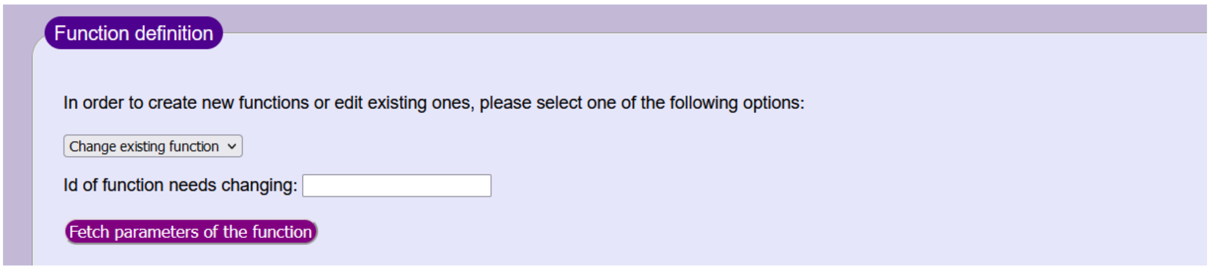


```
  id:                                                                                      "urn:ngsi-ld:test:Alert:count5"
  type:                                                                                    "https://uri.fiware.org/ns/data-models#Alert"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/category:
    type:                                                                                  "Property"
    value:                                                                                 "generated_data"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/subCategory:
    type:                                                                                  "Property"
    value:                                                                                 "alert data"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/validTo:
    type:                                                                                  "Property"
  ▼ value:
      @type:                                                                               "DateTime"
      @value:                                                                              "2023-05-02T13:32:04.541711+0Z"
▼ description:
    type:                                                                                  "Property"
    value:                                                                                 "Entity generated/updated"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/dateIssued:
    type:                                                                                  "Property"
  ▼ value:
      @type:                                                                               "DateTime"
      @value:                                                                              "2023-05-02T13:32:04.541615+0Z"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/alertSource:
    type:                                                                                  "Relationship"
    object:                                                                                "urn:ngsi-ld:dcf-logic-engine"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/source:
    type:                                                                                  "Property"
    value:                                                                                 "DCF"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/validFrom:
    type:                                                                                  "Property"
  ▼ value:
      @type:                                                                               "DateTime"
      @value:                                                                              "2023-05-02T13:32:04.541673+0Z"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/severity:
    type:                                                                                  "Property"
    value:                                                                                 "informational"
▼ humanVerified:
    type:                                                                                  "Property"
    value:                                                                                 "true"
▼ value:
    type:                                                                                  "Property"
    value:                                                                                 7
    observedAt:                                                                            "2023-05-02T13:32:04.541Z"
```

The operation is registered, and output is published.

Modifying existing operation

To modifying existing operation, select “Change existing function” option.



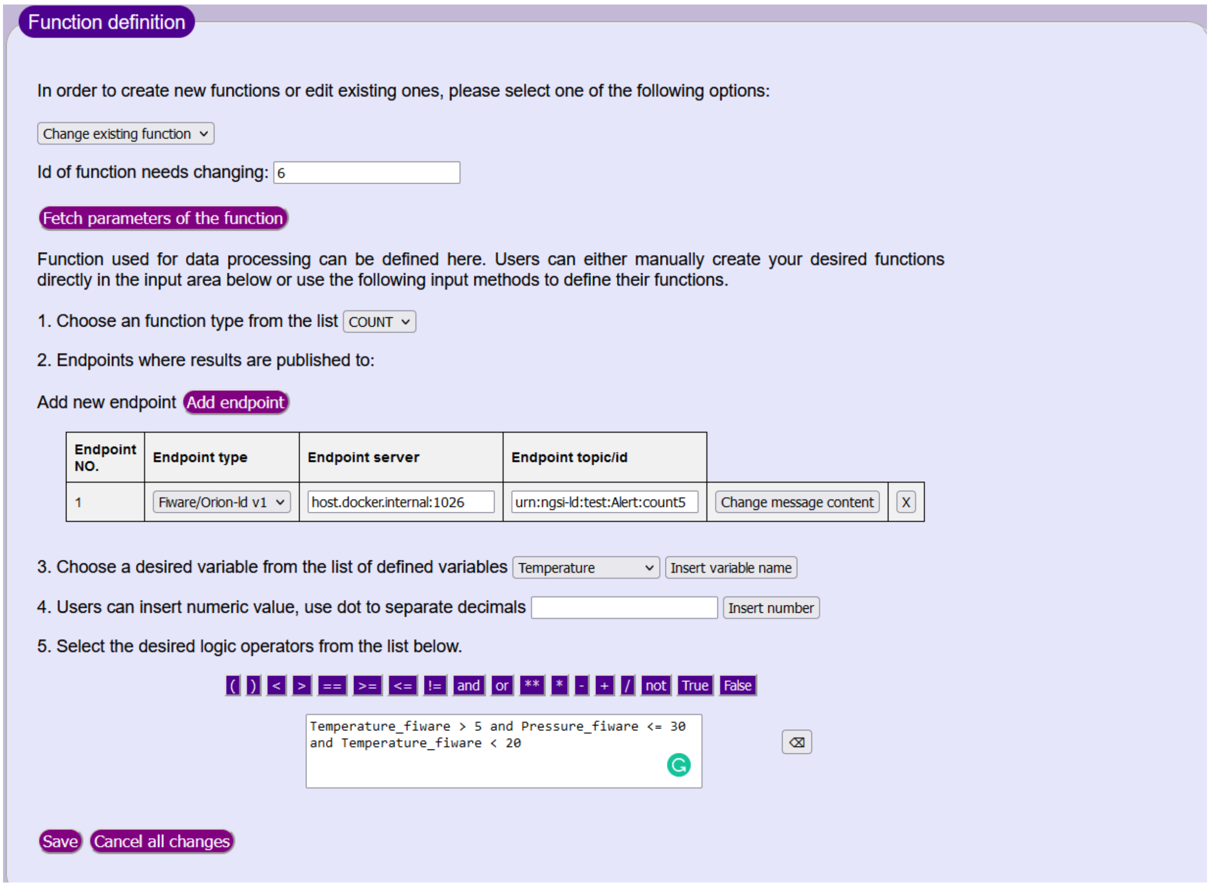
In order to create new functions or edit existing ones, please select one of the following options:

Change existing function

Id of function needs changing:

From here, users need to provide the id of the operation that needs to be modified, for example, the operation last created with id 6 needs to add new logic condition, after providing the id, “Fetch the operation parameters” button need to be clicked to retrieve the operation parameter.

New condition is added



In order to create new functions or edit existing ones, please select one of the following options:

Change existing function

Id of function needs changing: 6

Function used for data processing can be defined here. Users can either manually create your desired functions directly in the input area below or use the following input methods to define their functions.

1. Choose an function type from the list COUNT

2. Endpoints where results are published to:

Add new endpoint

Endpoint NO.	Endpoint type	Endpoint server	Endpoint topic/id		
1	Fiware/Orion-ld v1	host.docker.internal:1026	urn:ngsi-ld:test:Alert:count5	Change message content	X

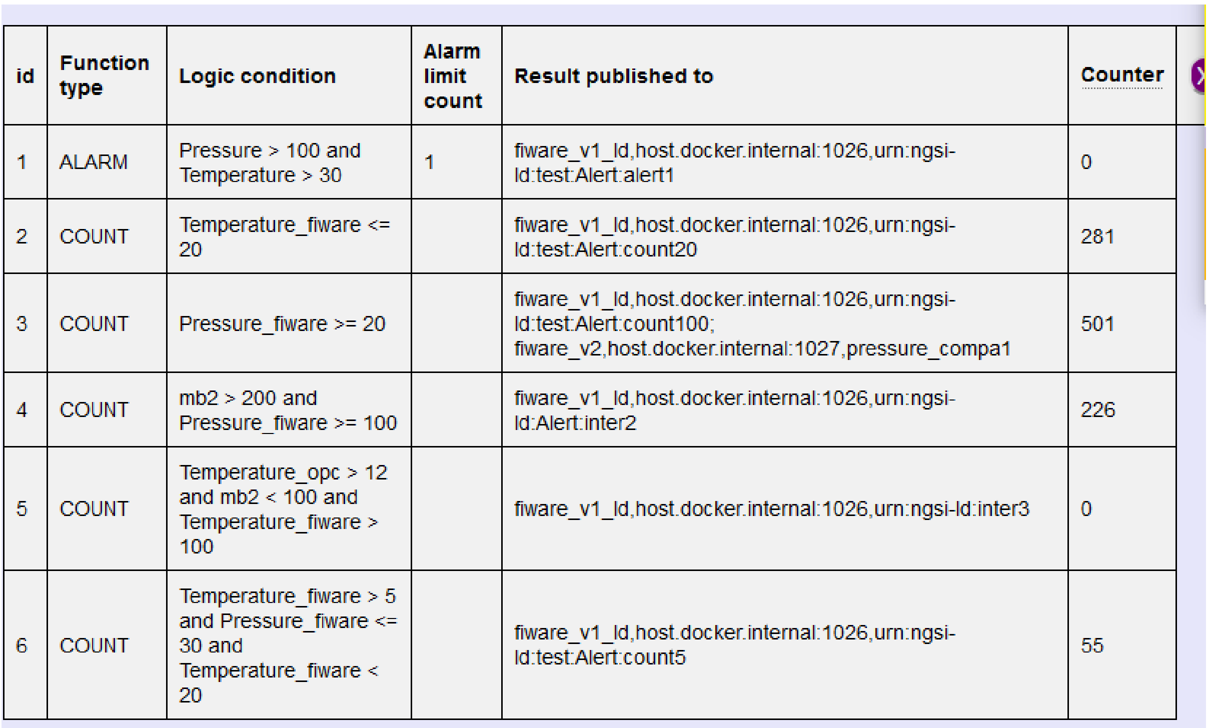
3. Choose a desired variable from the list of defined variables Temperature

4. Users can insert numeric value, use dot to separate decimals

5. Select the desired logic operators from the list below.

```
Temperature_fiware > 5 and Pressure_fiware <= 30
and Temperature_fiware < 20
```

After modifying the logic condition, the change can be saved or cancelled. The new operation saved will inherit the counter data from the original function.

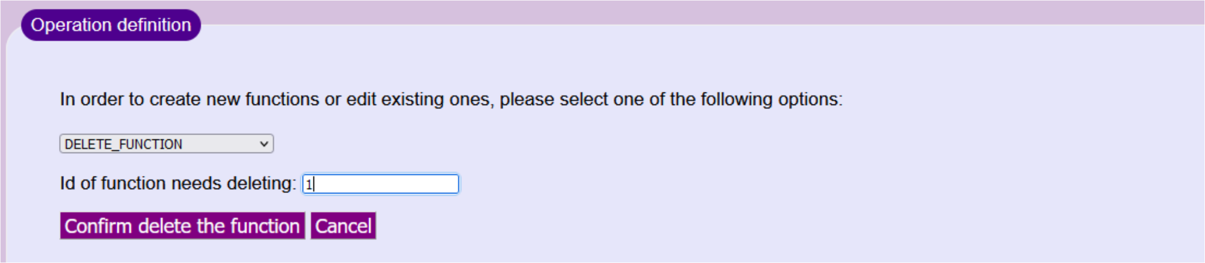


id	Function type	Logic condition	Alarm limit count	Result published to	Counter
1	ALARM	Pressure > 100 and Temperature > 30	1	fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:alert1	0
2	COUNT	Temperature_fiware <= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count20	281
3	COUNT	Pressure_fiware >= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count100; fiware_v2,host.docker.internal:1027,pressure_compa1	501
4	COUNT	mb2 > 200 and Pressure_fiware >= 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:Alert:inter2	226
5	COUNT	Temperature_opc > 12 and mb2 < 100 and Temperature_fiware > 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:inter3	0
6	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30 and Temperature_fiware < 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count5	55

Deleting existing operation

To delete existing function, choose “DELETE_FUNCTION” option

In this section, users need to provide the id of the operation that needs to be deleted, for example in this case operation with id 1 with logic condition “Pressure > 100 and Temperature > 30”



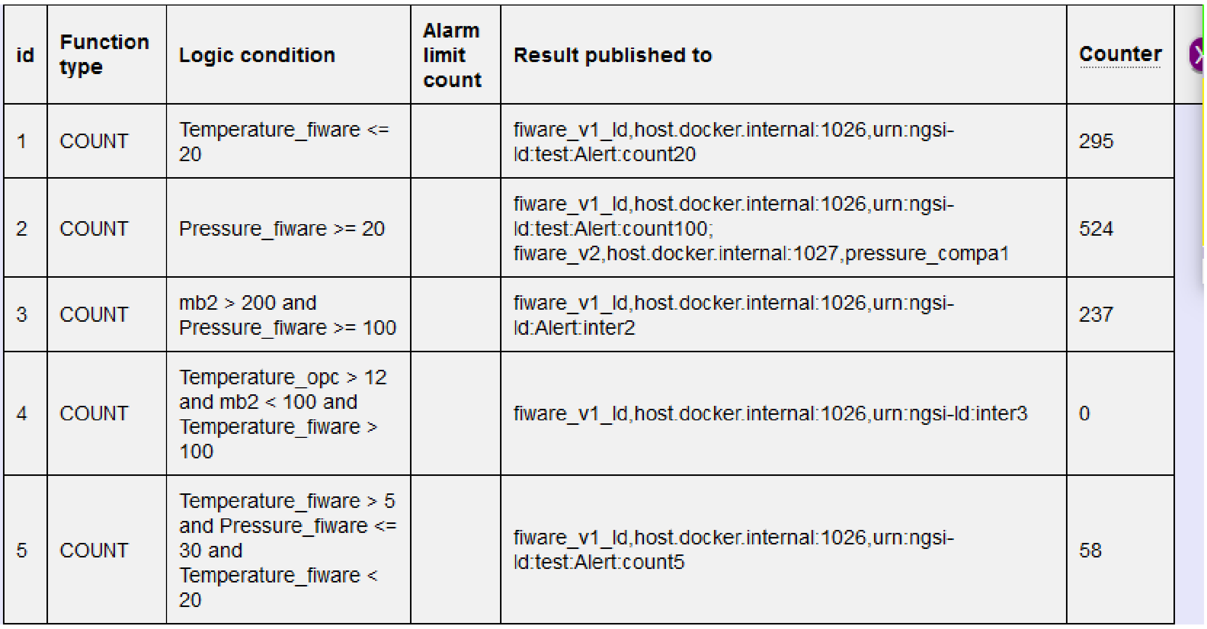
Operation definition

In order to create new functions or edit existing ones, please select one of the following options:

DELETE_FUNCTION

Id of function needs deleting: 1

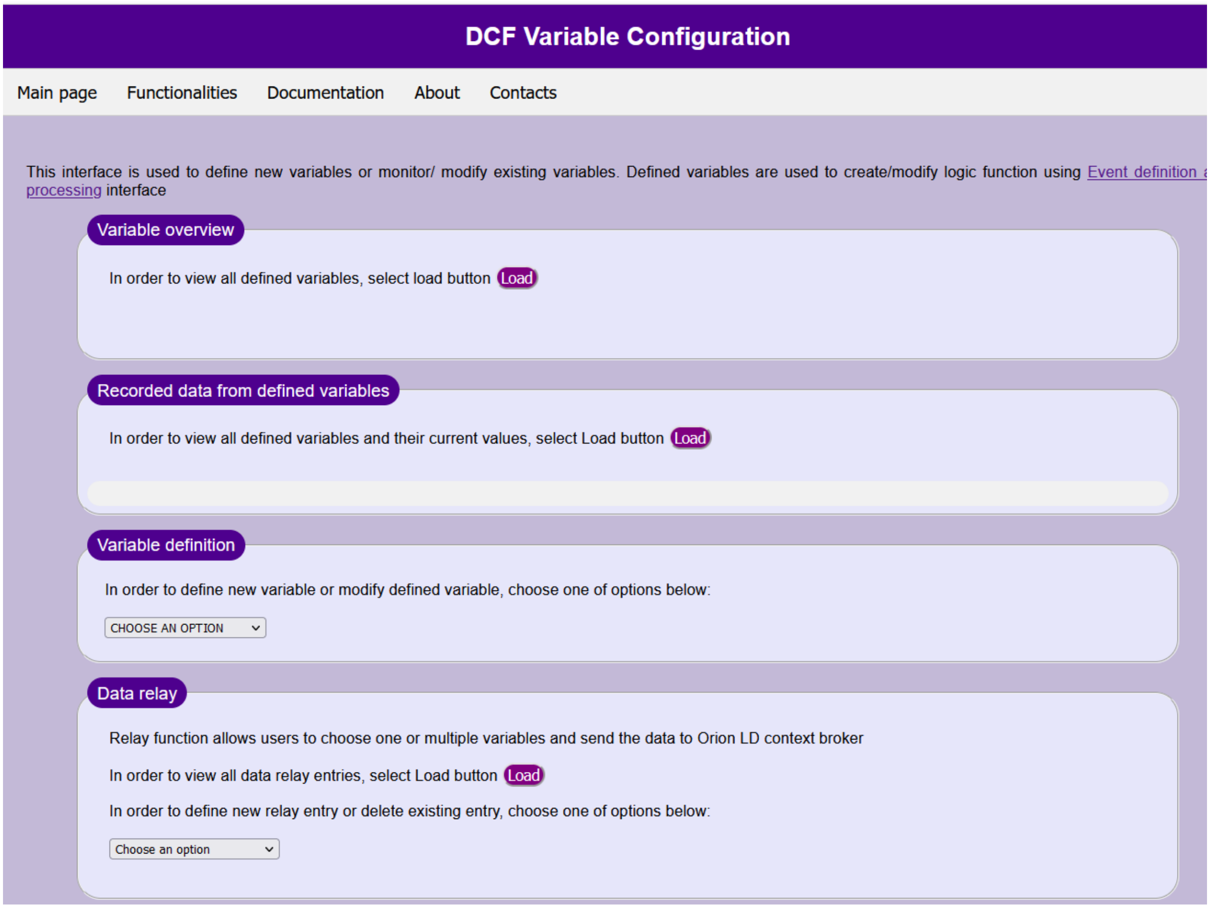
Click “confirm delete the function” to delete the selection operation, once committed, the count record of the function is deleted as well, so this needs to be done carefully



id	Function type	Logic condition	Alarm limit count	Result published to	Counter
1	COUNT	Temperature_fiware <= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count20	295
2	COUNT	Pressure_fiware >= 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count100; fiware_v2,host.docker.internal:1027,pressure_compa1	524
3	COUNT	mb2 > 200 and Pressure_fiware >= 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:Alert:inter2	237
4	COUNT	Temperature_opc > 12 and mb2 < 100 and Temperature_fiware > 100		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:inter3	0
5	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30 and Temperature_fiware < 20		fiware_v1_ld,host.docker.internal:1026,urn:ngsi-ld:test:Alert:count5	58

Input parameters interface

Input parameters interface is used to define, delete or modify the input variables that are used for logic operations. The interface also allows users to plot the data in live mode (maximum 50 data points) or data saved within the integrated Mongo-DB database



This interface is used to define new variables or monitor/ modify existing variables. Defined variables are used to create/modify logic function using Event definition a processing interface

Variable overview

In order to view all defined variables, select load button Load

Recorded data from defined variables

In order to view all defined variables and their current values, select Load button Load

Variable definition

In order to define new variable or modify defined variable, choose one of options below:

Data relay

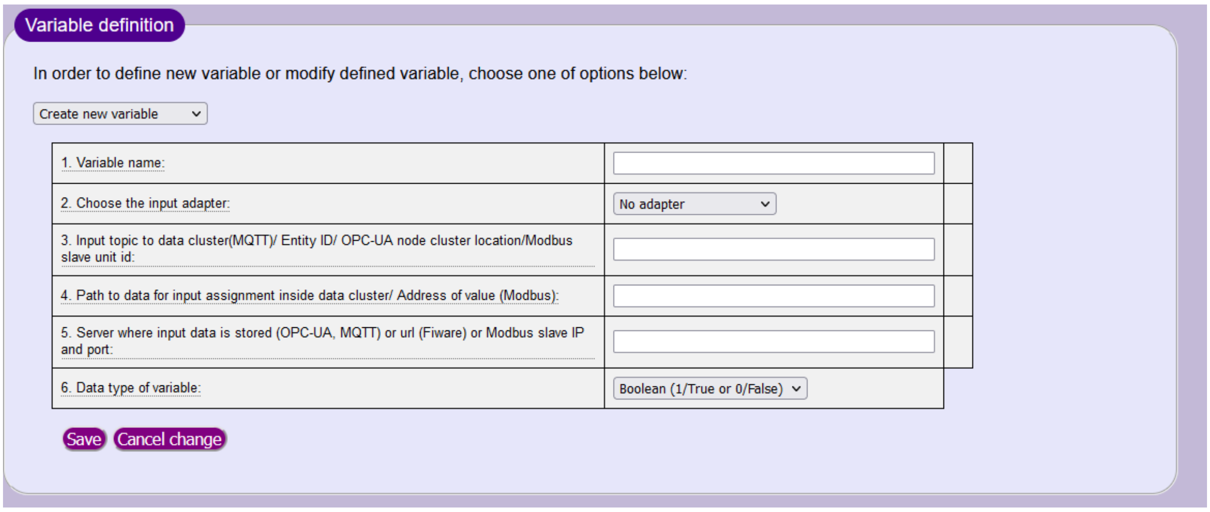
Relay function allows users to choose one or multiple variables and send the data to Orion LD context broker

In order to view all data relay entries, select Load button Load

In order to define new relay entry or delete existing entry, choose one of options below:

Defining new input variables

Choose option “Create new variable” to bring up new section



Variable definition

In order to define new variable or modify defined variable, choose one of options below:

Create new variable

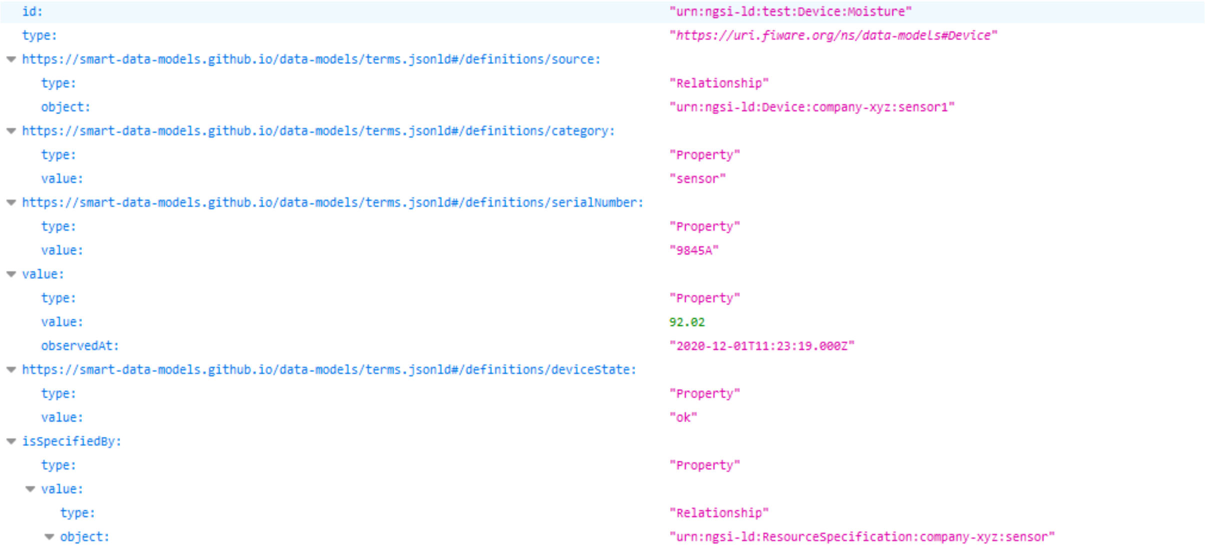
1. Variable name:	
2. Choose the input adapter:	No adapter
3. Input topic to data cluster(MQTT)/ Entity ID/ OPC-UA node cluster location/Modbus slave unit id:	
4. Path to data for input assignment inside data cluster/ Address of value (Modbus):	
5. Server where input data is stored (OPC-UA, MQTT) or url (Fiware) or Modbus slave IP and port:	
6. Data type of variable:	Boolean (1/True or 0/False)

List of parameters:

- Input name: the name of input variable to be assigned value to

- Adapter choice: the protocol that carries the input
- Input topic to data cluster (MQTT)/ Entity ID/ OPC-UA node cluster location: depends on the adapter type, this can be either topic (MQTT), entity id (fiware), node id (OPC-UA), where the data cluster (json file or data) is stored.
- Path to data for input assignment inside data cluster: path within the json file where relevant data can be extracted (if json is used), can be empty if data can be used directly
- Server where input data is stored (OPC-UA, MQTT) or url (Fiware): the server/link where the data is located.
- Input data type.

For example, the value for Moisture_fiware variable is located in dockerized fiware orion-ld server with entity id urn:ngsi-ld:test:Device:Moisture, the json file has format:



```
  id:                                                                       "urn:ngsi-ld:test:Device:Moisture"
  type:                                                                     "https://uri.fiware.org/ns/data-models#Device"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/source:
    type:                                                                   "Relationship"
    object:                                                                 "urn:ngsi-ld:Device:company-xyz:sensor1"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/category:
    type:                                                                   "Property"
    value:                                                                  "sensor"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/serialNumber:
    type:                                                                   "Property"
    value:                                                                  "9845A"
▼ value:
    type:                                                                   "Property"
    value:                                                                  92.02
    observedAt:                                                             "2020-12-01T11:23:19.000Z"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/deviceState:
    type:                                                                   "Property"
    value:                                                                  "ok"
▼ isSpecifiedBy:
    type:                                                                   "Property"
  ▼ value:
      type:                                                                 "Relationship"
    ▼ object:                                                               "urn:ngsi-ld:ResourceSpecification:company-xyz:sensor"
```

And users want to extract only the numeric value of the entity, so the path to the value is "value/value". Thus, the inputs for the variable is

Input name: Moisture_fiware

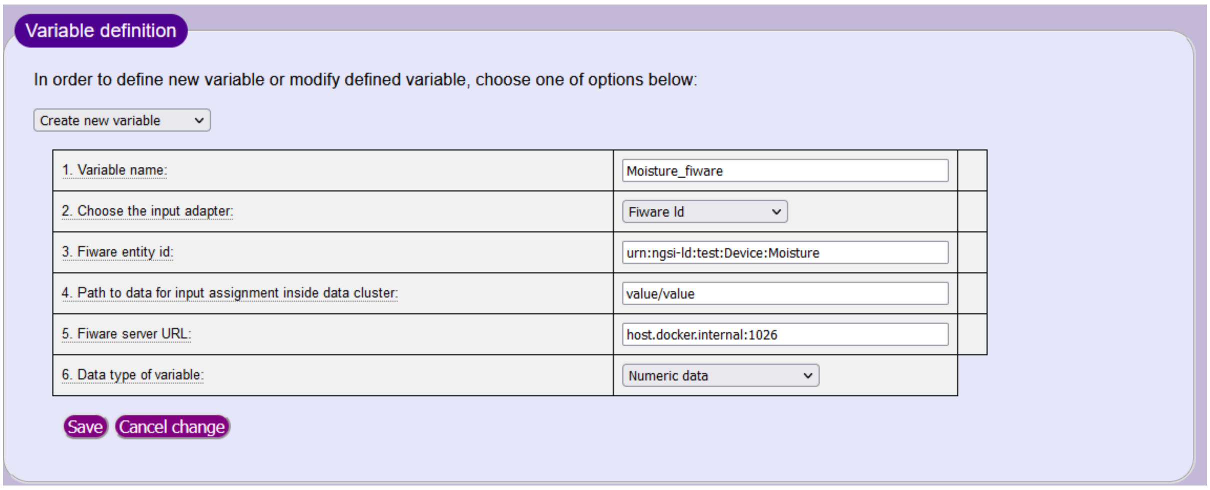
Choose your adapter: Fiware ld

Input topic to data cluster(MQTT)/ Entity ID/ OPC-UA node cluster location: urn:ngsi-ld:test:Device:Moisture

Path to data for input assignment inside data cluster: value/value

Server where input data is stored (OPC-UA, MQTT) or url (Fiware): host.docker.internal:1026

Data type: numeric data

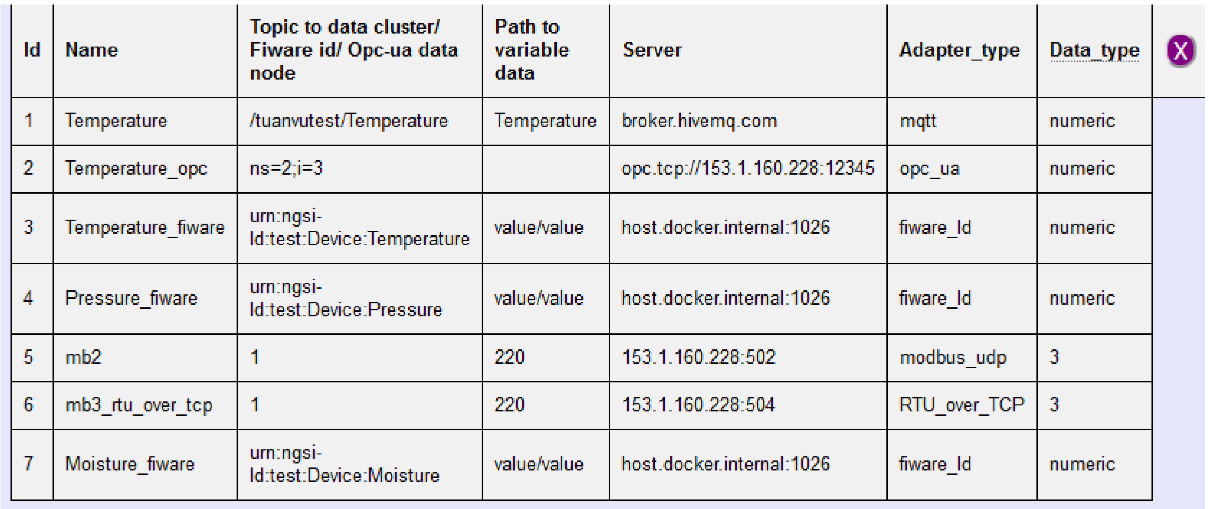


In order to define new variable or modify defined variable, choose one of options below:

Create new variable

1. Variable name:	Moisture_fiware
2. Choose the input adapter:	Fiware Id
3. Fiware entity id:	urn:ngsi-ld:test:Device:Moisture
4. Path to data for input assignment inside data cluster:	value/value
5. Fiware server URL:	host.docker.internal:1026
6. Data type of variable:	Numeric data

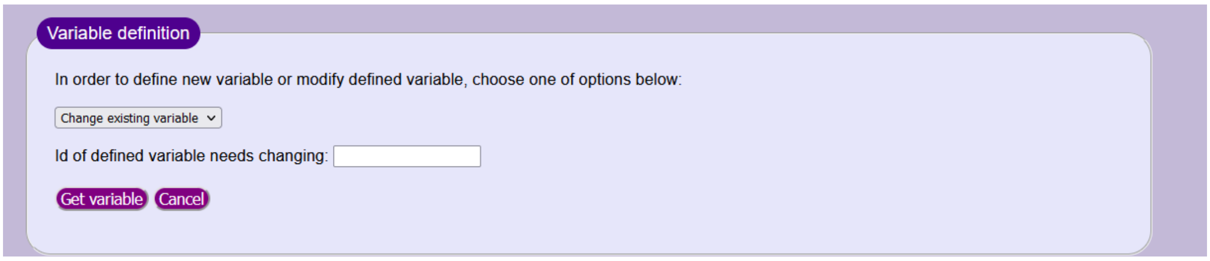
To save the variable, choose “Save”, or else, choose “Cancel change”



Id	Name	Topic to data cluster/ Fiware id/ Opc-ua data node	Path to variable data	Server	Adapter_type	Data_type	X
1	Temperature	/tuanvutest/Temperature	Temperature	broker.hivemq.com	mqtt	numeric	
2	Temperature_opc	ns=2;i=3		opc.tcp://153.1.160.228:12345	opc_ua	numeric	
3	Temperature_fiware	urn:ngsi-Id:test:Device:Temperature	value/value	host.docker.internal:1026	fiware_Id	numeric	
4	Pressure_fiware	urn:ngsi-Id:test:Device:Pressure	value/value	host.docker.internal:1026	fiware_Id	numeric	
5	mb2	1	220	153.1.160.228:502	modbus_udp	3	
6	mb3_rtu_over_tcp	1	220	153.1.160.228:504	RTU_over_TCP	3	
7	Moisture_fiware	urn:ngsi-Id:test:Device:Moisture	value/value	host.docker.internal:1026	fiware_Id	numeric	

Modify existing input

To modify existing variable, choose “change existing variable”

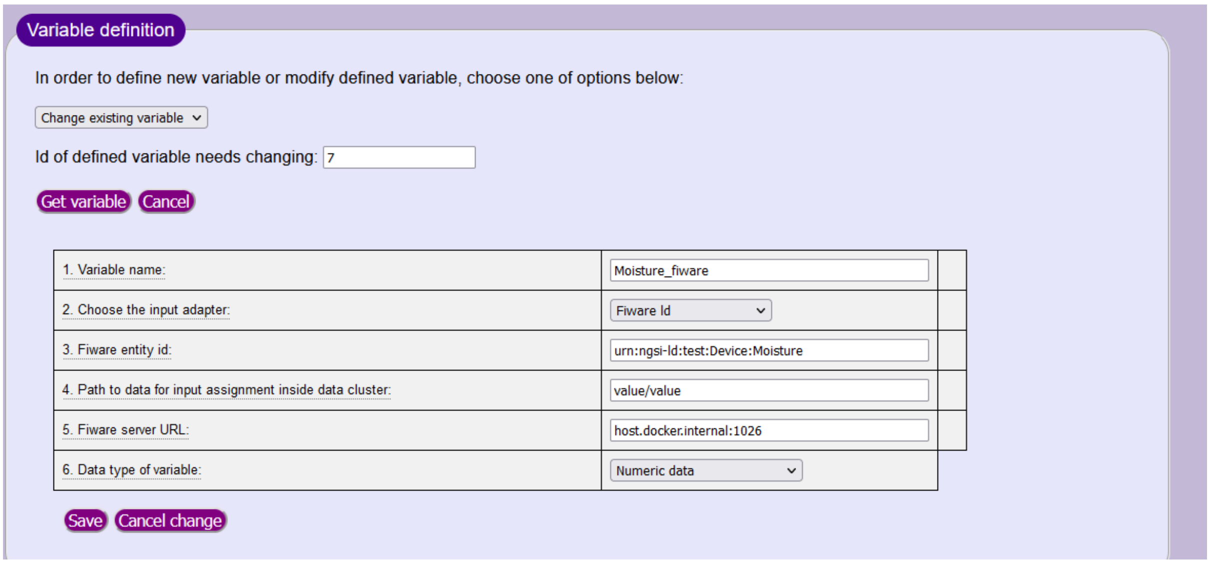


In order to define new variable or modify defined variable, choose one of options below:

Change existing variable

Id of defined variable needs changing:

The id of the variable need to be provided, after confirming the id, the input variable detail is loaded in the windows.



In order to define new variable or modify defined variable, choose one of options below:

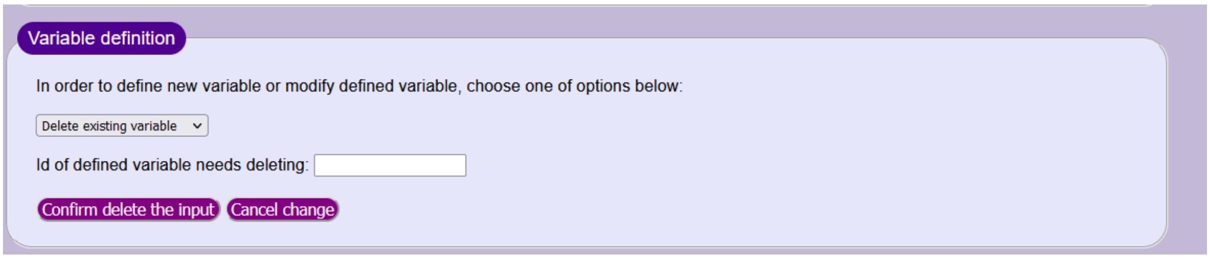
Change existing variable

Id of defined variable needs changing: 7

1. Variable name:	Moisture_fiware
2. Choose the input adapter:	Fiware Id
3. Fiware entity id:	urn:ngsi-ld:test:Device:Moisture
4. Path to data for input assignment inside data cluster:	value/value
5. Fiware server URL:	host.docker.internal:1026
6. Data type of variable:	Numeric data

Delete existing variable

To delete existing input variable, select "delete existing variable" and provide the id of the input needing to be deleted



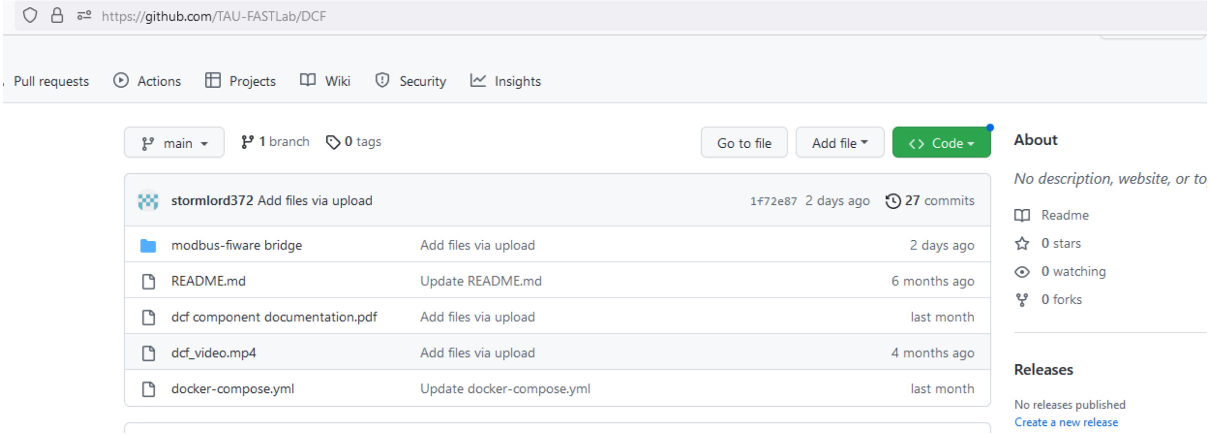
In order to define new variable or modify defined variable, choose one of options below:

Delete existing variable

Id of defined variable needs deleting:

Note for inputs that are from Modbus RTU and Modbus ASCII

In order to receive data from Modbus RTU and Modbus ASCII, additional script is needed since dockerized application cannot access host serial COM ports.



https://github.com/TAU-FASTLab/DCF

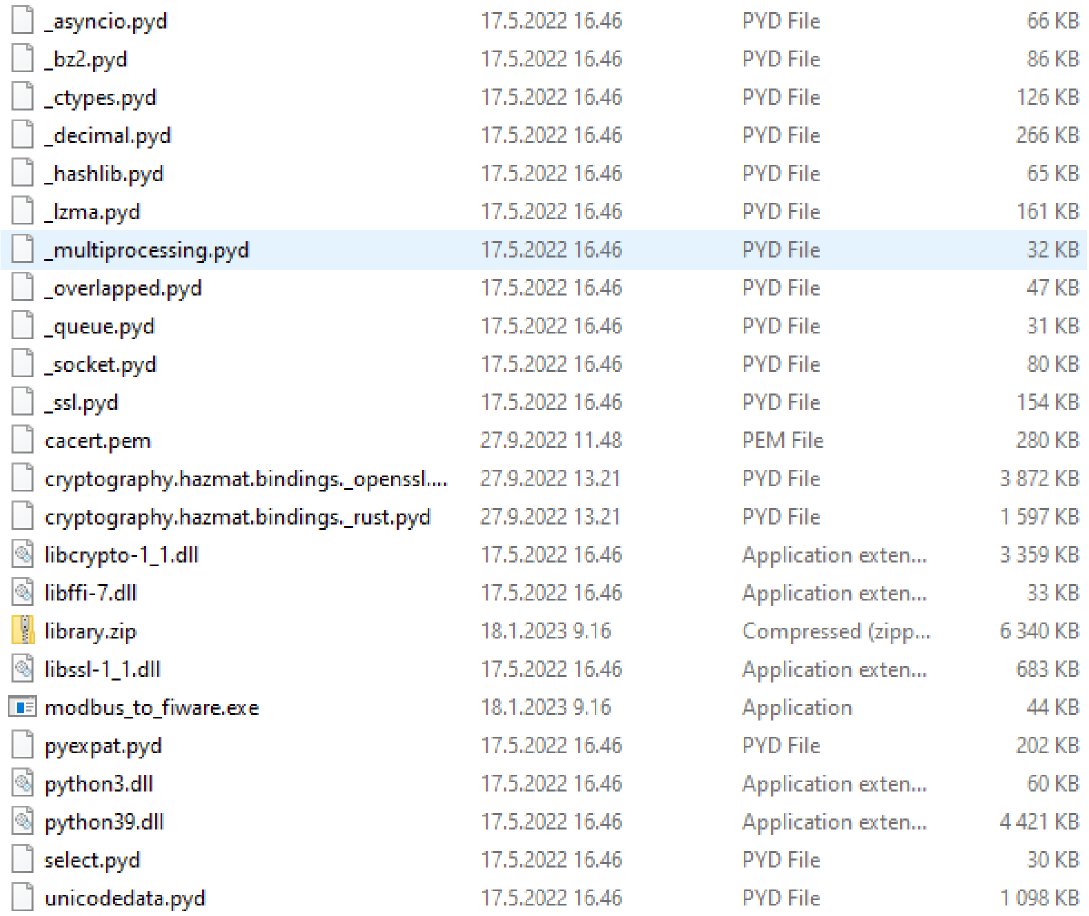
stormlord372 Add files via upload 1f72e87 2 days ago 27 commits

modbus-fiware bridge	Add files via upload	2 days ago
README.md	Update README.md	6 months ago
dcf component documentation.pdf	Add files via upload	last month
dcf_video.mp4	Add files via upload	4 months ago
docker-compose.yml	Update docker-compose.yml	last month

No description, website, or to

No releases published

modbus-fiware-bridge module can be downloaded from DCF repository.

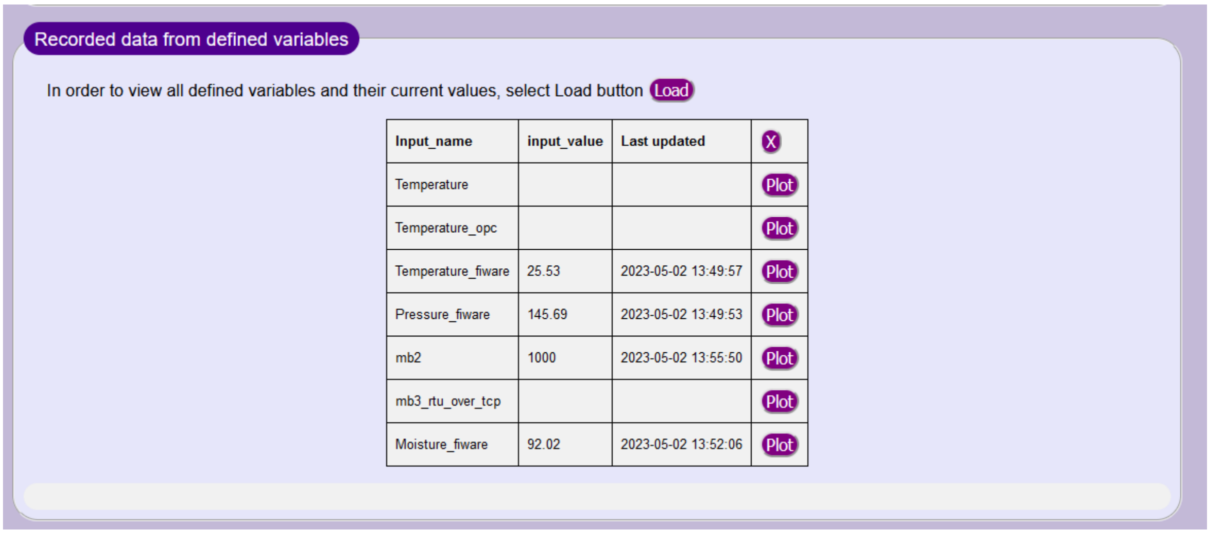


_asyncio.pyd	17.5.2022 16.46	PYD File	66 KB
_bz2.pyd	17.5.2022 16.46	PYD File	86 KB
_ctypes.pyd	17.5.2022 16.46	PYD File	126 KB
_decimal.pyd	17.5.2022 16.46	PYD File	266 KB
_hashlib.pyd	17.5.2022 16.46	PYD File	65 KB
_lzma.pyd	17.5.2022 16.46	PYD File	161 KB
_multiprocessing.pyd	17.5.2022 16.46	PYD File	32 KB
_overlapped.pyd	17.5.2022 16.46	PYD File	47 KB
_queue.pyd	17.5.2022 16.46	PYD File	31 KB
_socket.pyd	17.5.2022 16.46	PYD File	80 KB
_ssl.pyd	17.5.2022 16.46	PYD File	154 KB
cacert.pem	27.9.2022 11.48	PEM File	280 KB
cryptography.hazmat.bindings._openssl....	27.9.2022 13.21	PYD File	3 872 KB
cryptography.hazmat.bindings._rust.pyd	27.9.2022 13.21	PYD File	1 597 KB
libcrypto-1_1.dll	17.5.2022 16.46	Application exten...	3 359 KB
libffi-7.dll	17.5.2022 16.46	Application exten...	33 KB
library.zip	18.1.2023 9.16	Compressed (zipp...	6 340 KB
libssl-1_1.dll	17.5.2022 16.46	Application exten...	683 KB
modbus_to_fiware.exe	18.1.2023 9.16	Application	44 KB
pyexpat.pyd	17.5.2022 16.46	PYD File	202 KB
python3.dll	17.5.2022 16.46	Application exten...	60 KB
python39.dll	17.5.2022 16.46	Application exten...	4 421 KB
select.pyd	17.5.2022 16.46	PYD File	30 KB
unicodedata.pyd	17.5.2022 16.46	PYD File	1 098 KB

Once the module is downloaded, run modbus_to_fiware.exe. The program will receive input details, gather the data from modbus system and transfer the collected data to Fiware where DCF can read the data.

Retrieve and plot the data.

The interface also allows users to retrieve and plot data

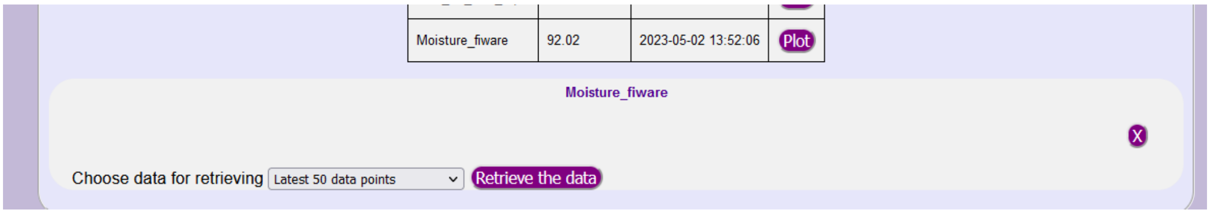


Recorded data from defined variables

In order to view all defined variables and their current values, select Load button

Input_name	input_value	Last updated	X
Temperature			Plot
Temperature_opc			Plot
Temperature_fiware	25.53	2023-05-02 13:49:57	Plot
Pressure_fiware	145.69	2023-05-02 13:49:53	Plot
mb2	1000	2023-05-02 13:55:50	Plot
mb3_rtu_over_tcp			Plot
Moisture_fiware	92.02	2023-05-02 13:52:06	Plot

Select the Plot button of the variable.



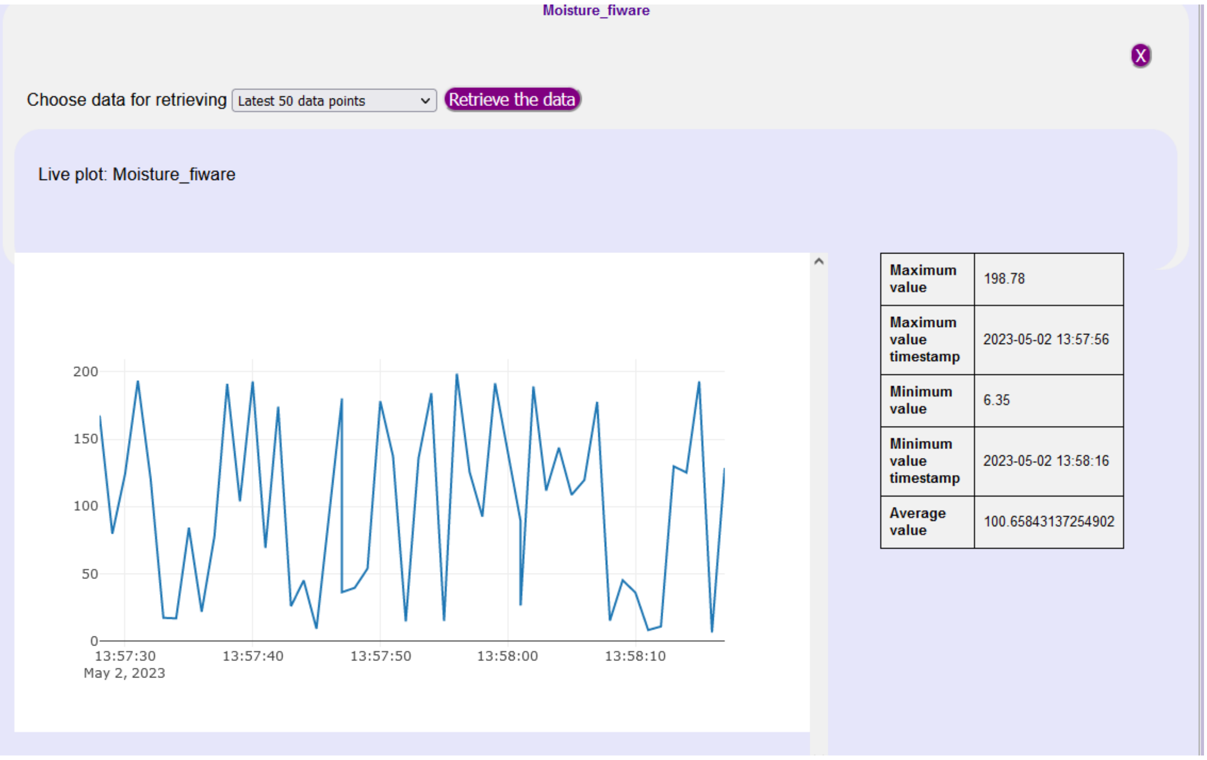
Moisture_fiware	92.02	2023-05-02 13:52:06	Plot

Moisture_fiware

Choose data for retrieving: Latest 50 data points

User can either plot latest 50 data points or retrieve data from database

Latest 50 data points



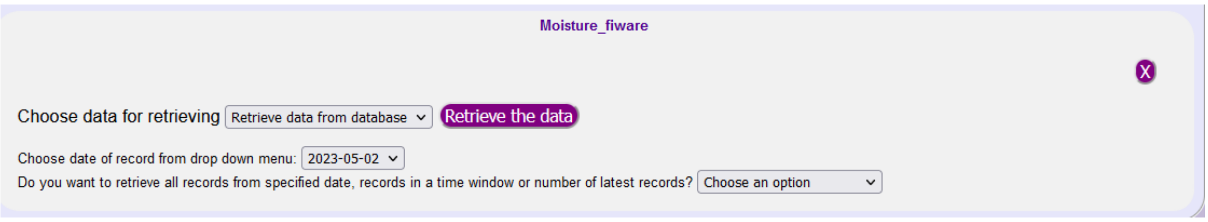
Moisture_fiware

Choose data for retrieving: Latest 50 data points

Live plot: Moisture_fiware

Maximum value	198.78
Maximum value timestamp	2023-05-02 13:57:56
Minimum value	6.35
Minimum value timestamp	2023-05-02 13:58:16
Average value	100.65843137254902

Retrieve data from database.

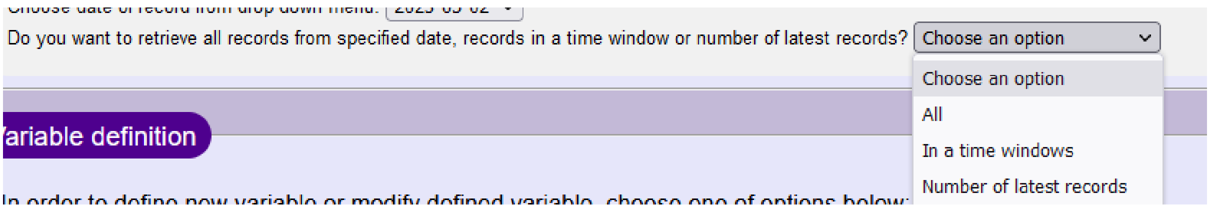


Choose data for retrieving: Retrieve data from database

Choose date of record from drop down menu: 2023-05-02

Do you want to retrieve all records from specified date, records in a time window or number of latest records? Choose an option

The list of dates of record and option to extract data can be seen.

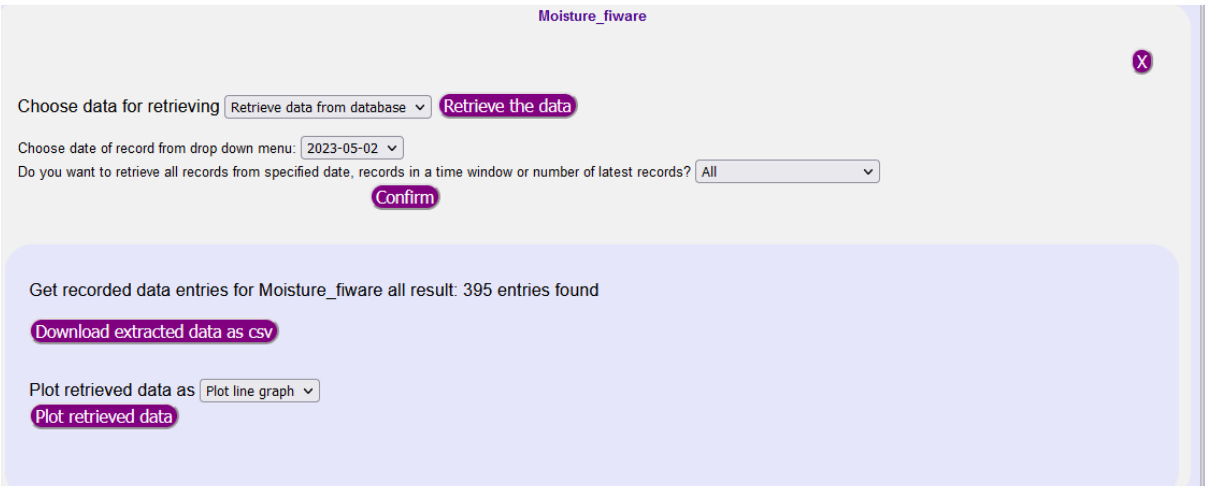


Do you want to retrieve all records from specified date, records in a time window or number of latest records? Choose an option

In order to define new variable or modify defined variable, choose one of options below:

- All: retrieve entire record from selected date
- In a time windows: retrieve record between 2 specified timestamps.
- Number of latest records: retrieve last records of the selected date with specific number for retrieving.

Retrieve all record from selected date example.



Choose data for retrieving: Retrieve data from database

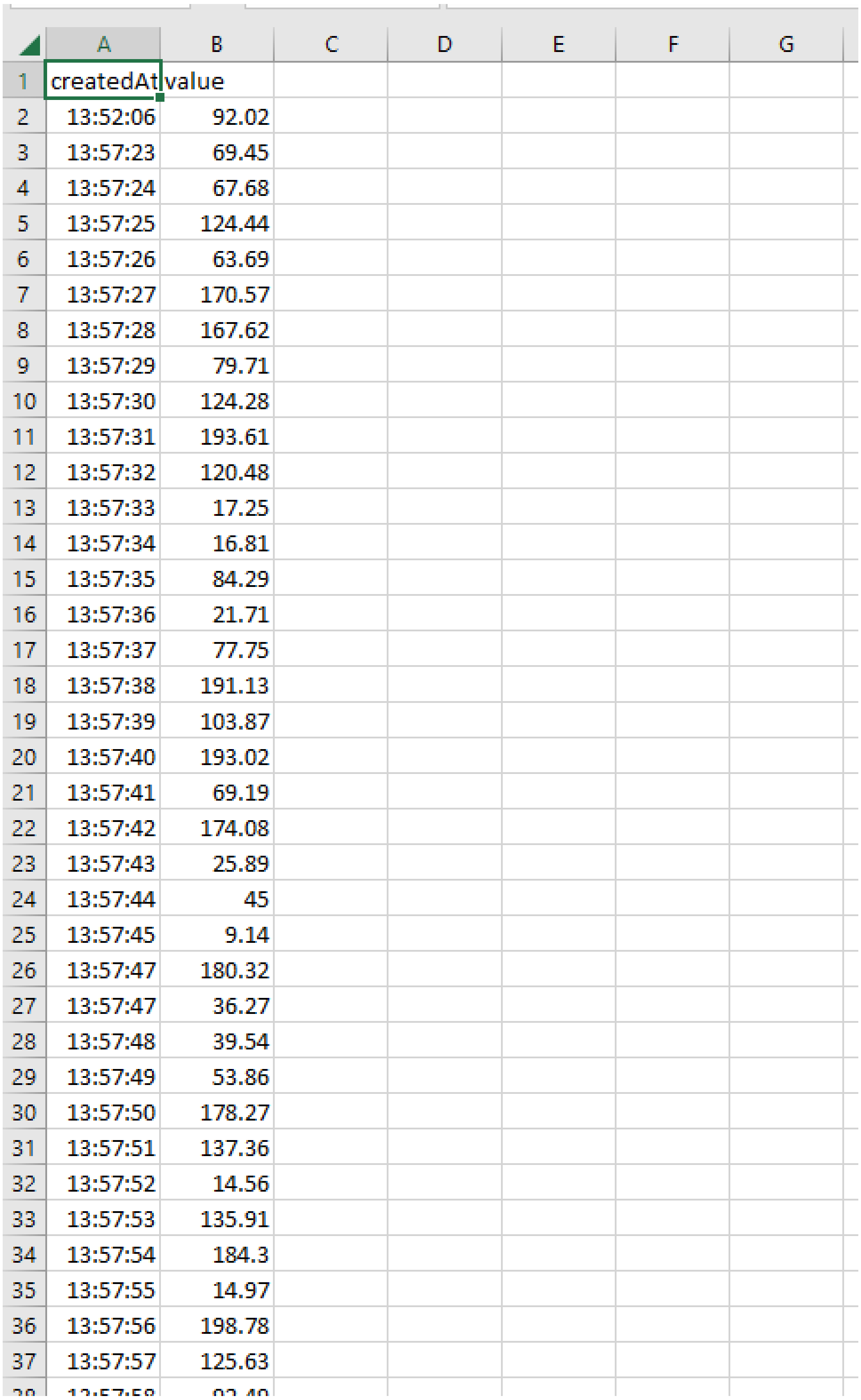
Choose date of record from drop down menu: 2023-05-02

Do you want to retrieve all records from specified date, records in a time window or number of latest records? All

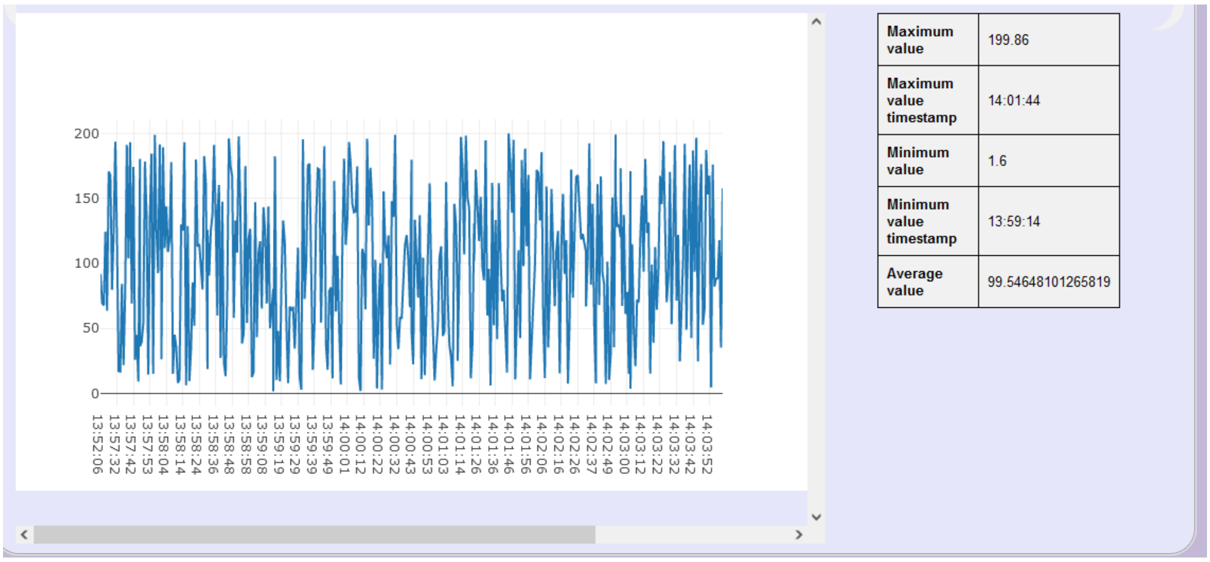
Get recorded data entries for Moisture_fiware all result: 395 entries found

Plot retrieved data as: Plot line graph

Users can download extracted data and/or plot the data



	A	B	C	D	E	F	G
1	createdAt	value					
2	13:52:06	92.02					
3	13:57:23	69.45					
4	13:57:24	67.68					
5	13:57:25	124.44					
6	13:57:26	63.69					
7	13:57:27	170.57					
8	13:57:28	167.62					
9	13:57:29	79.71					
10	13:57:30	124.28					
11	13:57:31	193.61					
12	13:57:32	120.48					
13	13:57:33	17.25					
14	13:57:34	16.81					
15	13:57:35	84.29					
16	13:57:36	21.71					
17	13:57:37	77.75					
18	13:57:38	191.13					
19	13:57:39	103.87					
20	13:57:40	193.02					
21	13:57:41	69.19					
22	13:57:42	174.08					
23	13:57:43	25.89					
24	13:57:44	45					
25	13:57:45	9.14					
26	13:57:47	180.32					
27	13:57:47	36.27					
28	13:57:48	39.54					
29	13:57:49	53.86					
30	13:57:50	178.27					
31	13:57:51	137.36					
32	13:57:52	14.56					
33	13:57:53	135.91					
34	13:57:54	184.3					
35	13:57:55	14.97					
36	13:57:56	198.78					
37	13:57:57	125.63					
38	13:57:58	92.40					

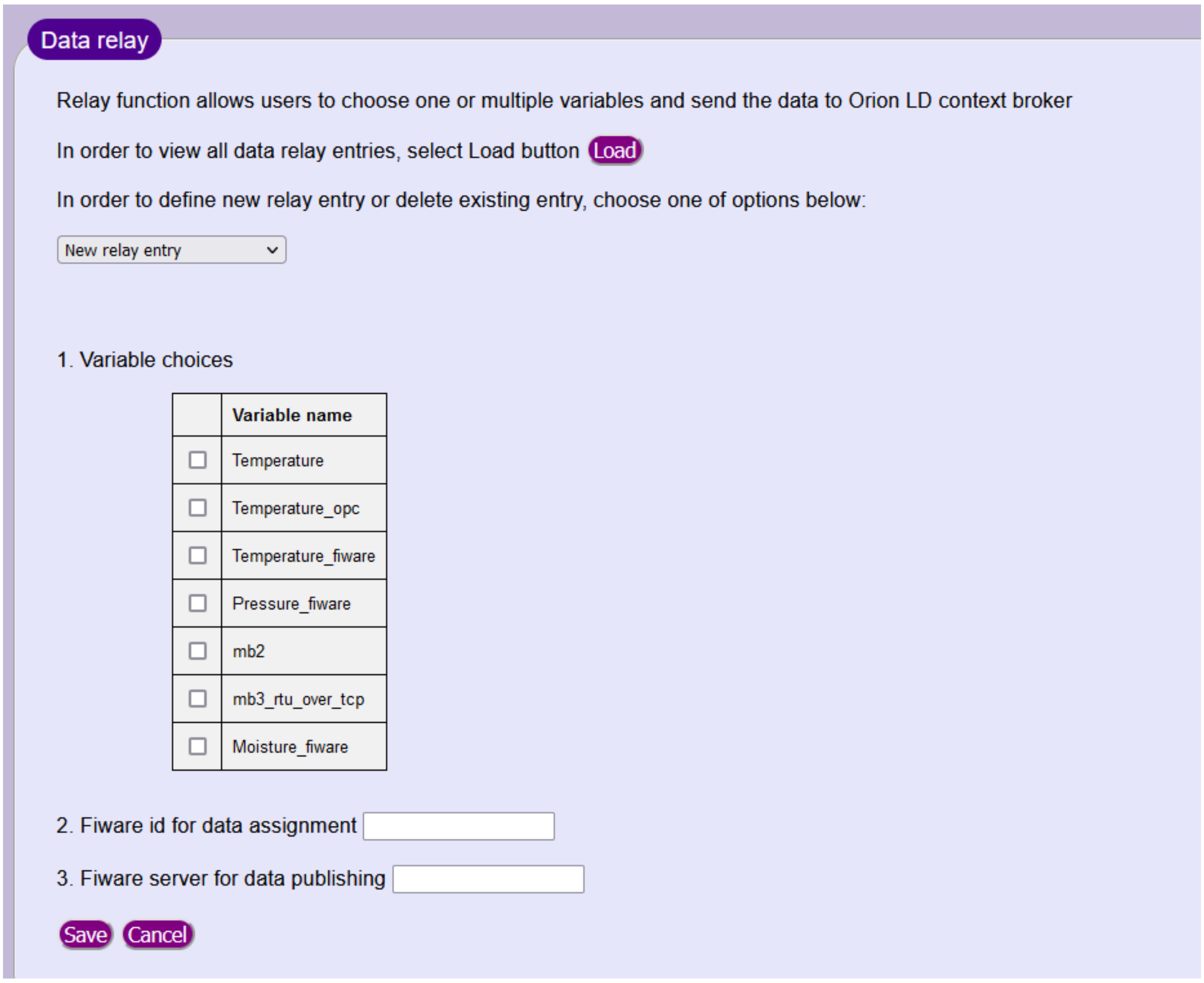


Maximum value	199.86
Maximum value timestamp	14:01:44
Minimum value	1.6
Minimum value timestamp	13:59:14
Average value	99.54648101265819

Rudimentary data analysis can be performed, showing maximum, minimum, and average values.

Data relay

The interface can collect data from several sources to create entities which can be published to Fiware Orion-LD broker.



Relay function allows users to choose one or multiple variables and send the data to Orion LD context broker

In order to view all data relay entries, select Load button

In order to define new relay entry or delete existing entry, choose one of options below:

New relay entry

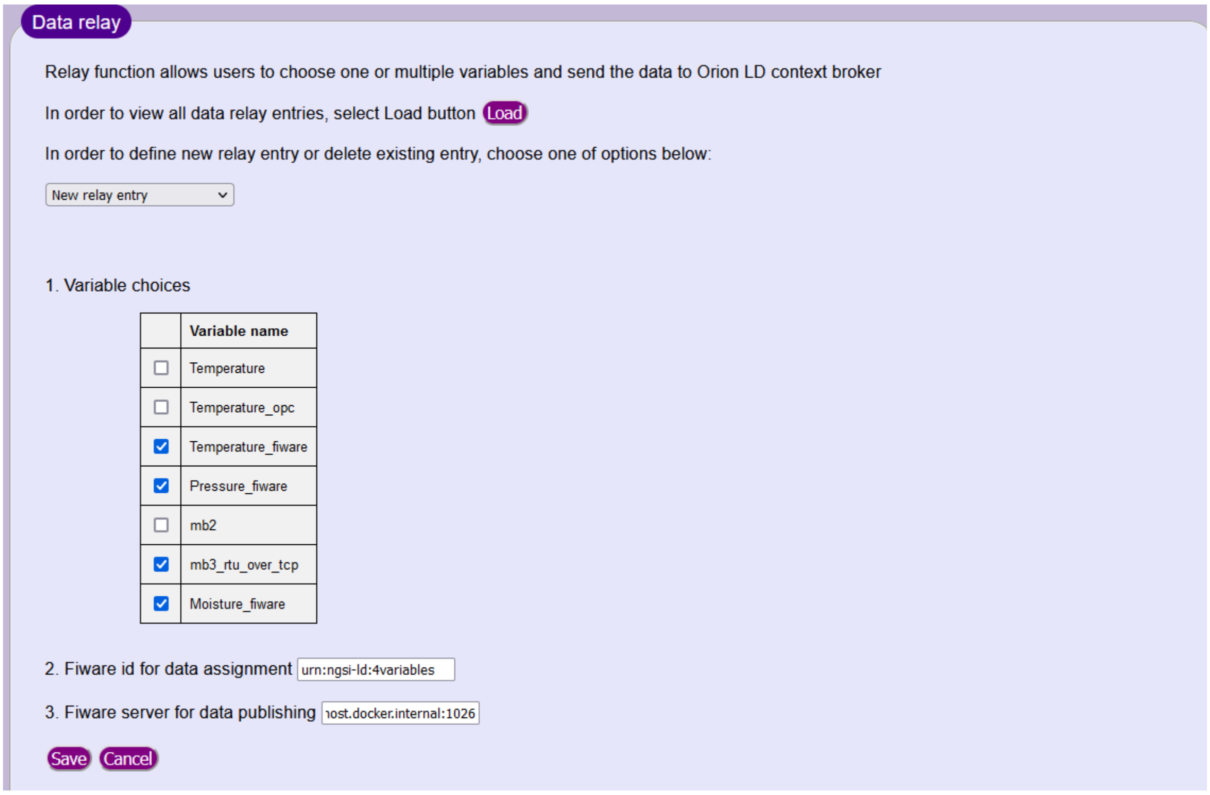
1. Variable choices

	Variable name
□	Temperature
□	Temperature_opc
□	Temperature_fiware
□	Pressure_fiware
□	mb2
□	mb3_rtu_over_tcp
□	Moisture_fiware

2. Fiware id for data assignment

3. Fiware server for data publishing

Example relay entry



Relay function allows users to choose one or multiple variables and send the data to Orion LD context broker

In order to view all data relay entries, select Load button

In order to define new relay entry or delete existing entry, choose one of options below:

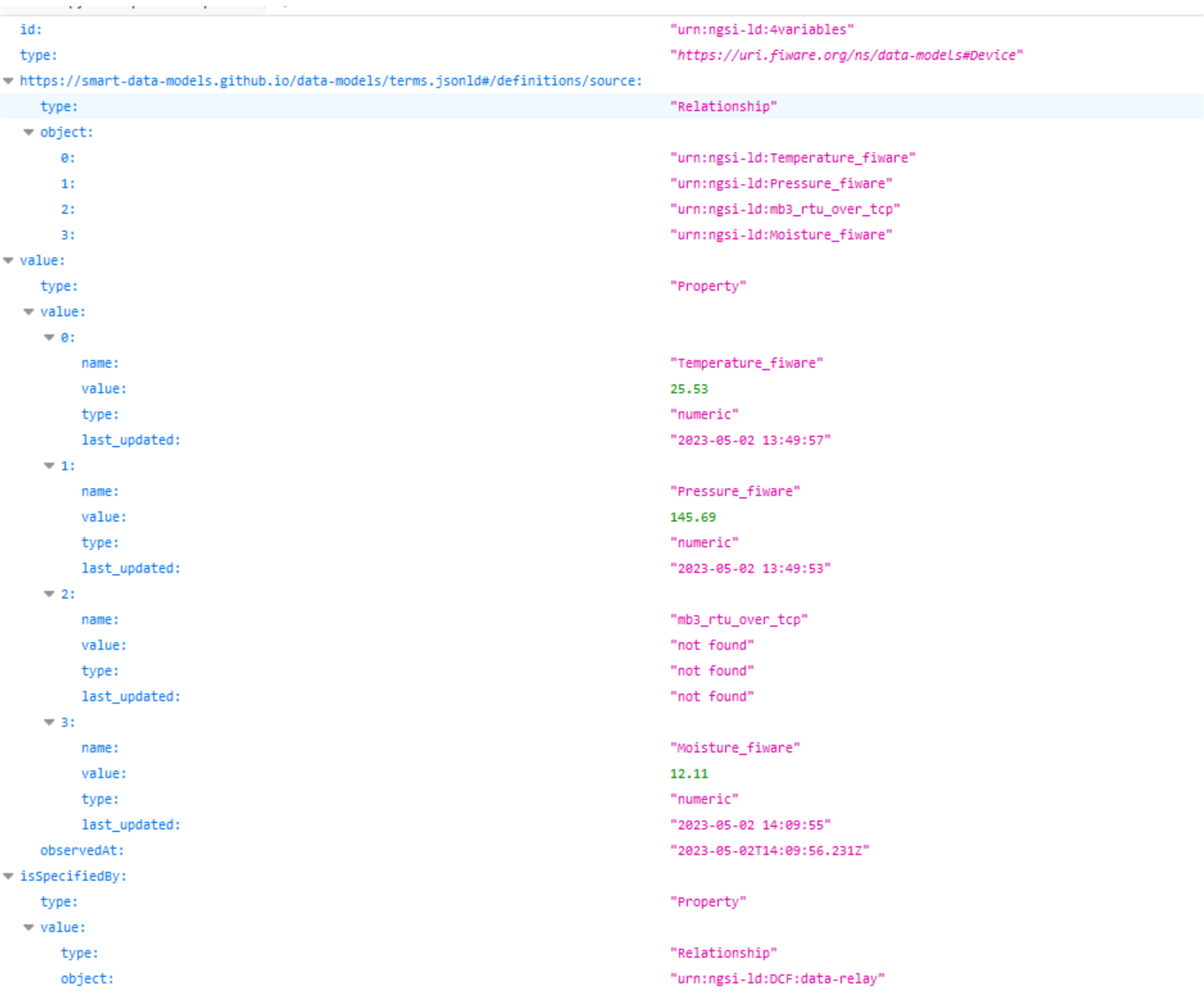
New relay entry

1. Variable choices

	Variable name
□	Temperature
□	Temperature_opc
☑	Temperature_fiware
☑	Pressure_fiware
□	mb2
☑	mb3_rtu_over_tcp
☑	Moisture_fiware

2. Fiware id for data assignment urn:ngsi-ld:4variables

3. Fiware server for data publishing ıost.docker.internal:1026

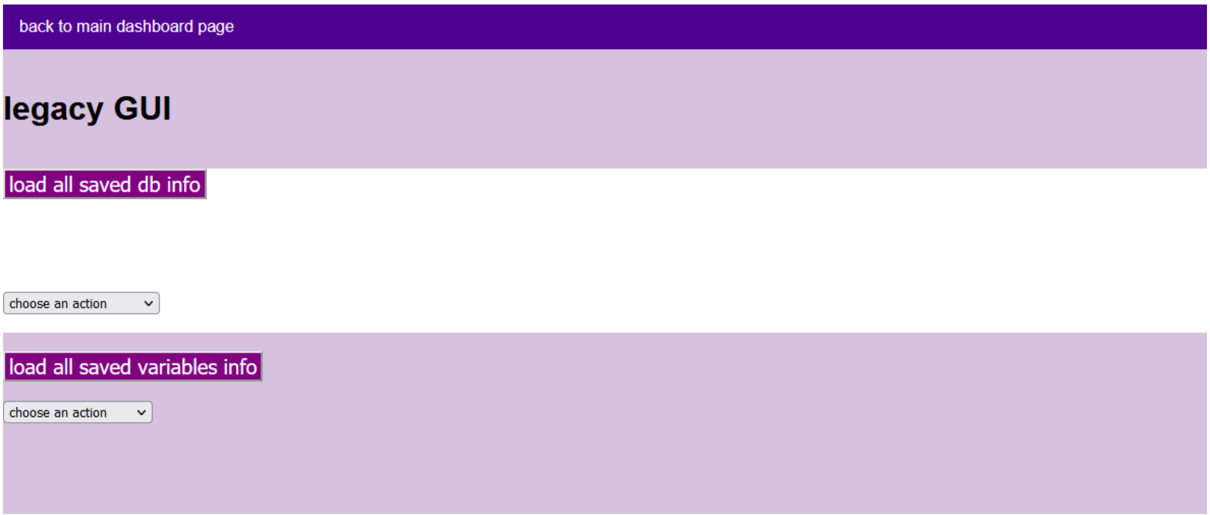


```
  id:                                                                          "urn:ngsi-ld:4variables"
  type:                                                                        "https://uri.fiware.org/ns/data-models#Device"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/source:
    type:                                                                      "Relationship"
  ▼ object:
      0:                                                                       "urn:ngsi-ld:Temperature_fiware"
      1:                                                                       "urn:ngsi-ld:Pressure_fiware"
      2:                                                                       "urn:ngsi-ld:mb3_rtu_over_tcp"
      3:                                                                       "urn:ngsi-ld:Moisture_fiware"
▼ value:
    type:                                                                      "Property"
  ▼ value:
    ▼ 0:
        name:                                                                  "Temperature_fiware"
        value:                                                                 25.53
        type:                                                                  "numeric"
        last_updated:                                                          "2023-05-02 13:49:57"
    ▼ 1:
        name:                                                                  "Pressure_fiware"
        value:                                                                 145.69
        type:                                                                  "numeric"
        last_updated:                                                          "2023-05-02 13:49:53"
    ▼ 2:
        name:                                                                  "mb3_rtu_over_tcp"
        value:                                                                 "not found"
        type:                                                                  "not found"
        last_updated:                                                          "not found"
    ▼ 3:
        name:                                                                  "Moisture_fiware"
        value:                                                                 12.11
        type:                                                                  "numeric"
        last_updated:                                                          "2023-05-02 14:09:55"
    observedAt:                                                                "2023-05-02T14:09:56.231Z"
▼ isSpecifiedBy:
    type:                                                                      "Property"
  ▼ value:
      type:                                                                    "Relationship"
      object:                                                                  "urn:ngsi-ld:DCF:data-relay"
```

Legacy data interface

The interface allows users to

- Define database entry with parameters for saving the data within DCF (applies for oracle and sap hana)
- Define fiware entities for data extraction from saved database (the same functionality is shared with mongoDB interface)

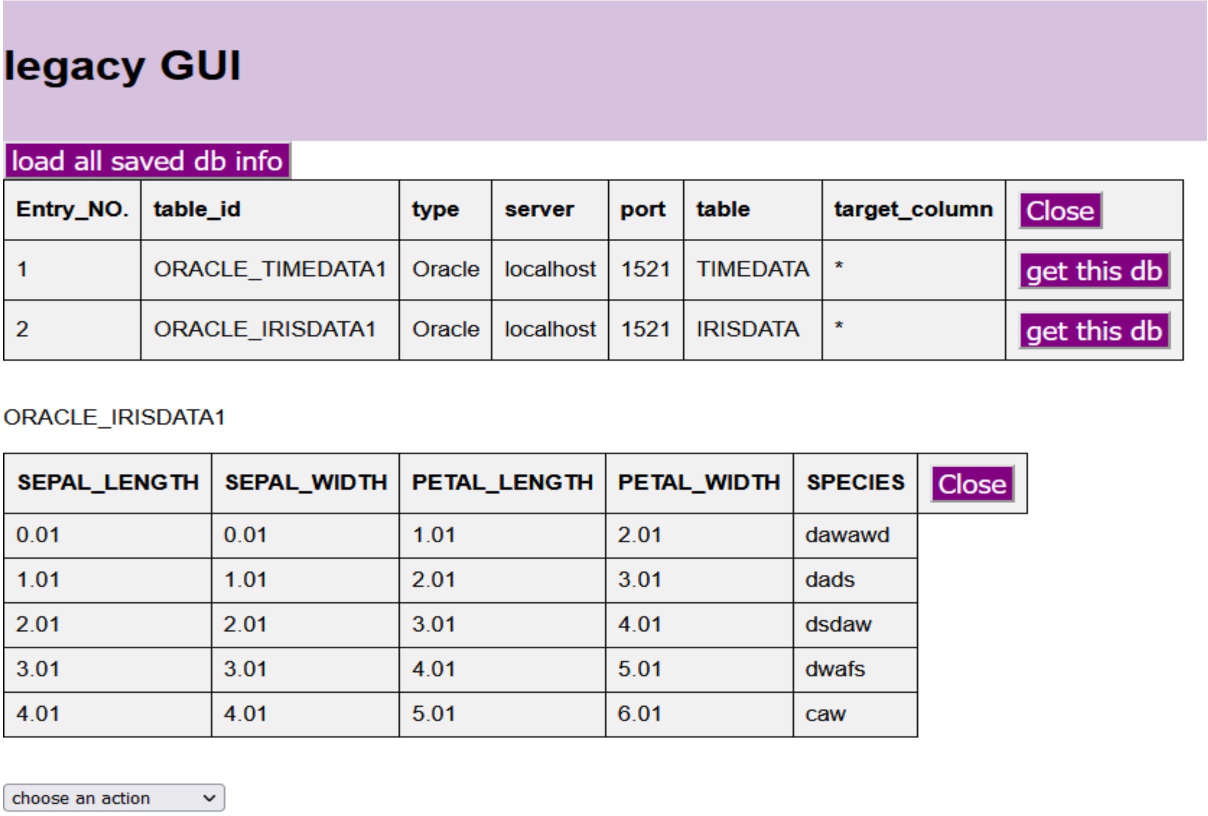


Database entry section

Displaying db table info and content

Allows users to monitor/add new/ delete existing database information and view table content if data is retrieved, if the data is not retrieved, the old entry needs to be deleted and defined again.

For example, saved data with table_id ORACLE_IRISDATA1 is displayed



Entry_NO.	table_id	type	server	port	table	target_column	Close
1	ORACLE_TIMEDATA1	Oracle	localhost	1521	TIMEDATA	*	get this db
2	ORACLE_IRISDATA1	Oracle	localhost	1521	IRISDATA	*	get this db

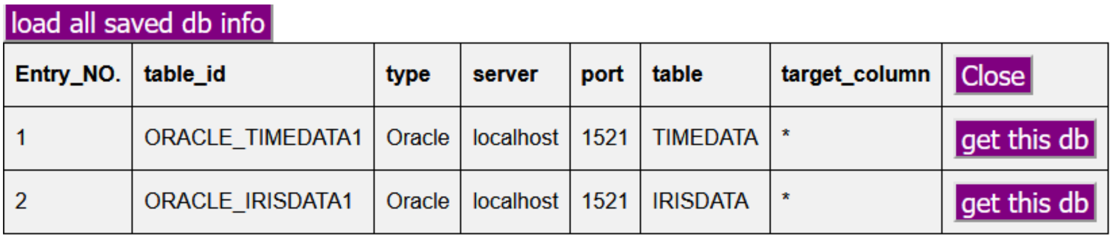
ORACLE_IRISDATA1

SEPAL_LENGTH	SEPAL_WIDTH	PETAL_LENGTH	PETAL_WIDTH	SPECIES	Close
0.01	0.01	1.01	2.01	dawawd	
1.01	1.01	2.01	3.01	dads	
2.01	2.01	3.01	4.01	dsdaw	
3.01	3.01	4.01	5.01	dwafs	
4.01	4.01	5.01	6.01	caw	

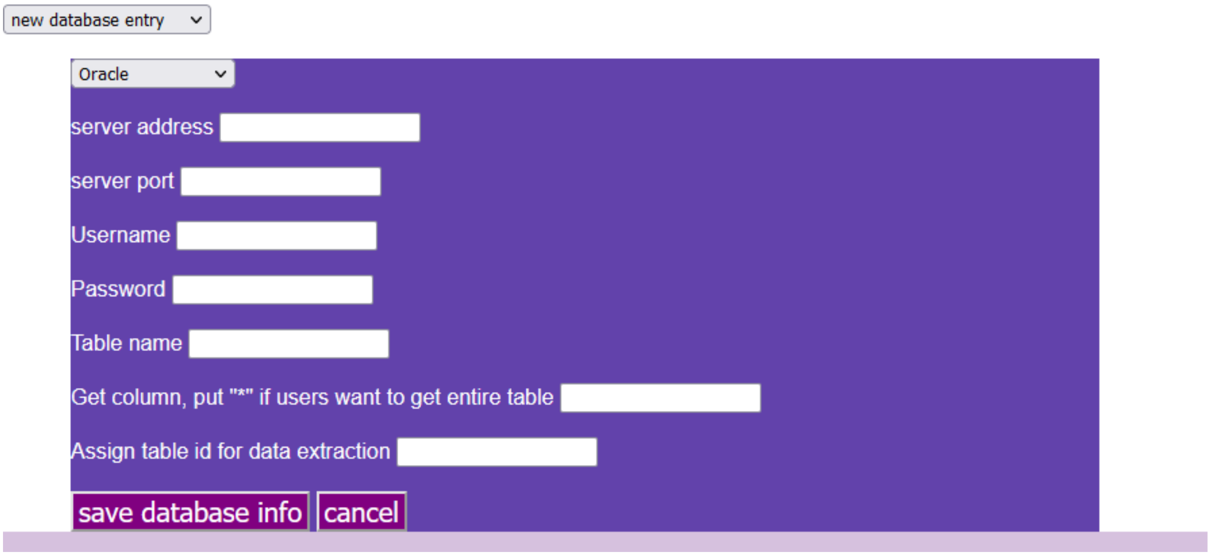
Adding new database entry and deleting existing entry

Adding new entry

To add new database entry, select "new database entry" to bring up new input section



Entry_NO.	table_id	type	server	port	table	target_column	Close
1	ORACLE_TIMEDATA1	Oracle	localhost	1521	TIMEDATA	*	get this db
2	ORACLE_IRISDATA1	Oracle	localhost	1521	IRISDATA	*	get this db



server address

server port

Username

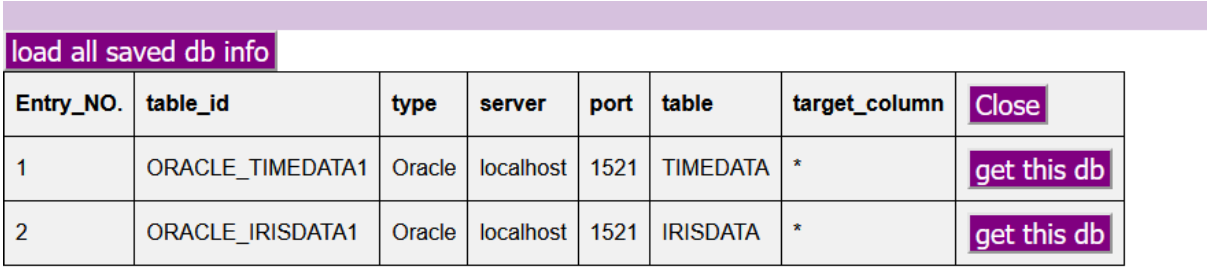
Password

Table name

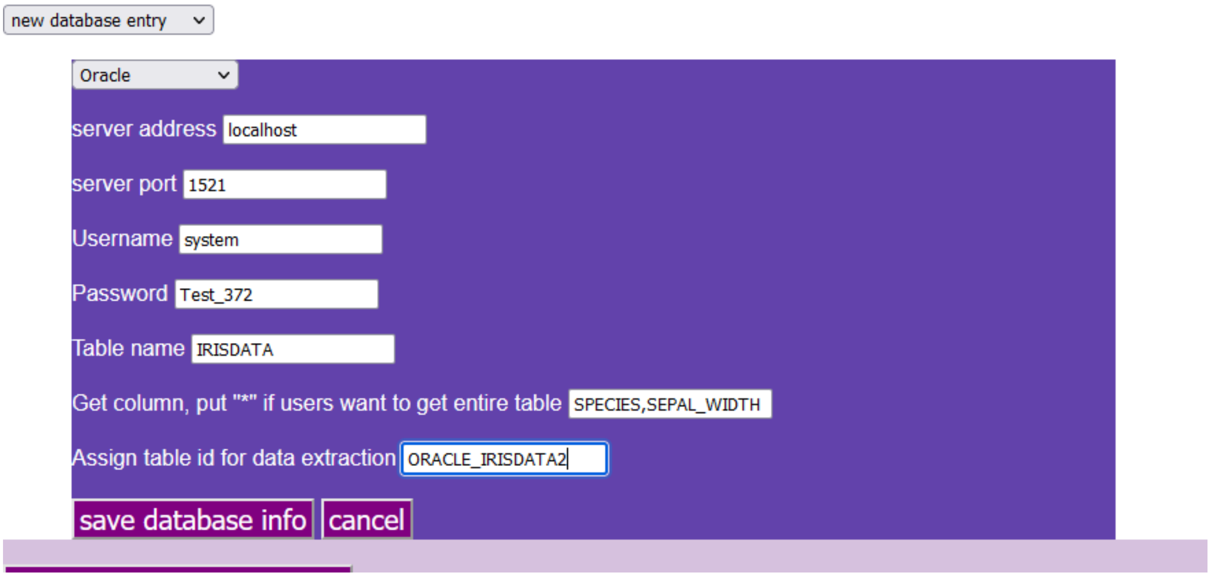
Get column, put "*" if users want to get entire table

Assign table id for data extraction

For example, data from column SPECIES and SEPAL_WIDTH of IRISDATA table from oracle database needs to be stored, to log in and retrieve data, username and password are also needed, the retrieved data is assigned to unique table_id ORACLE_IRISDATA2, the columns to be retrieved need to be separated by “,” (comma).



Entry_NO.	table_id	type	server	port	table	target_column	Close
1	ORACLE_TIMEDATA1	Oracle	localhost	1521	TIMEDATA	*	get this db
2	ORACLE_IRISDATA1	Oracle	localhost	1521	IRISDATA	*	get this db



server address localhost

server port 1521

Username system

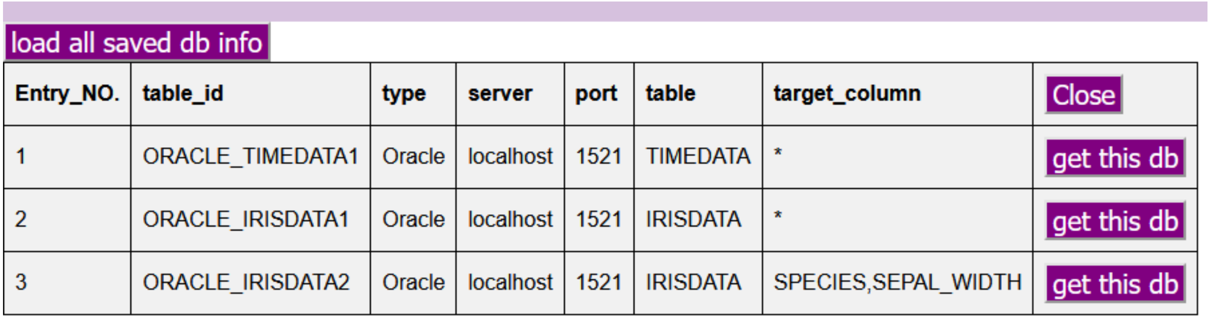
Password Test_372

Table name IRISDATA

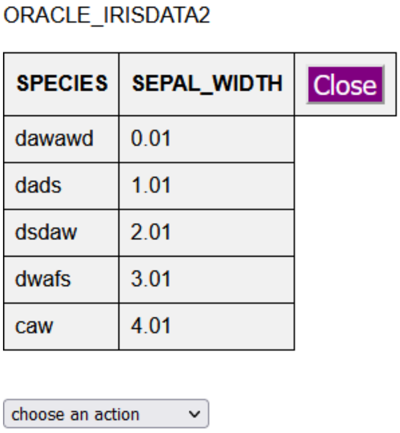
Get column, put "*" if users want to get entire table SPECIES,SEPAL_WIDTH

Assign table id for data extraction ORACLE_IRISDATA2

Click “save database info” and retrieve the data



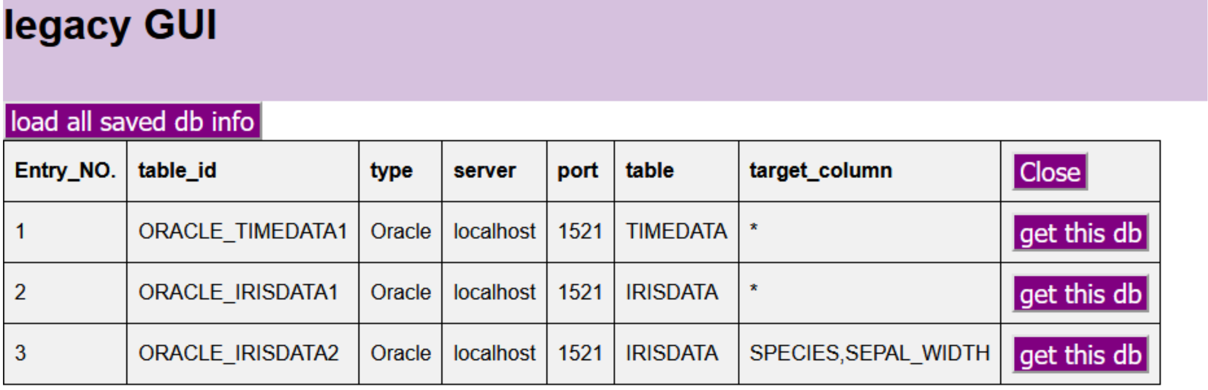
Entry_NO.	table_id	type	server	port	table	target_column	Close
1	ORACLE_TIMEDATA1	Oracle	localhost	1521	TIMEDATA	*	get this db
2	ORACLE_IRISDATA1	Oracle	localhost	1521	IRISDATA	*	get this db
3	ORACLE_IRISDATA2	Oracle	localhost	1521	IRISDATA	SPECIES,SEPAL_WIDTH	get this db


ORACLE_IRISDATA2

SPECIES	SEPAL_WIDTH	Close
dawawd	0.01	
dads	1.01	
dsdaw	2.01	
dwafs	3.01	
caw	4.01	

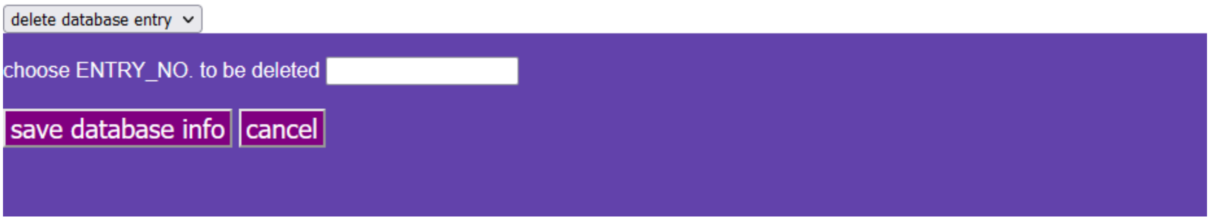
Deleting existing entry

To delete entry, choose “delete database entry” and provide the ENTRY_NO. of the needed entry



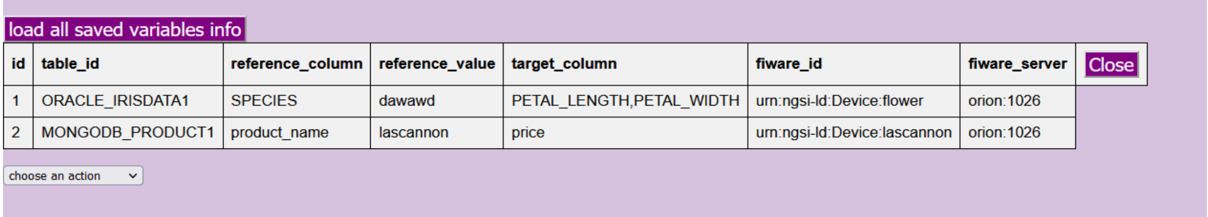
legacy GUI

Entry_NO.	table_id	type	server	port	table	target_column
1	ORACLE_TIMEDATA1	Oracle	localhost	1521	TIMEDATA	*
2	ORACLE_IRISDATA1	Oracle	localhost	1521	IRISDATA	*
3	ORACLE_IRISDATA2	Oracle	localhost	1521	IRISDATA	SPECIES,SEPAL_WIDTH



choose ENTRY_NO. to be deleted

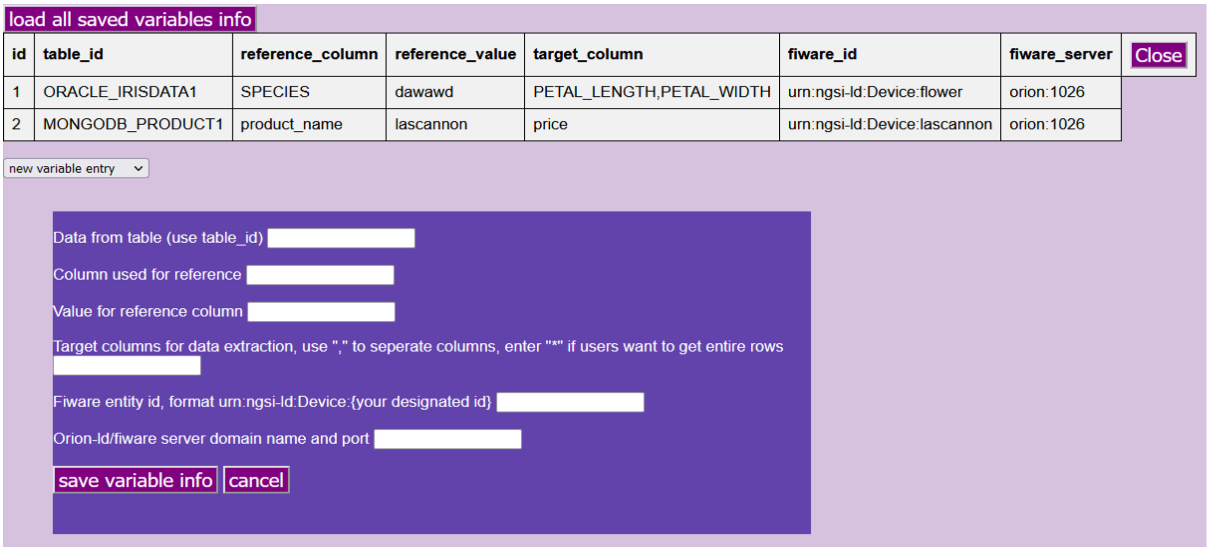
Adding new parameters for retrieving data from saved database entries or deleting existing entries



id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	orion:1026
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	orion:1026

Adding new entry for retrieving data

Data from saved database table can be extracted and sent through fiware as entity, this functionality is shared and can be shown in both MongoDB interface and legacy interface, to add new parameters entry, choose “new variable entry”



id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	orion:1026
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	orion:1026

Data from table (use table_id)

Column used for reference

Value for reference column

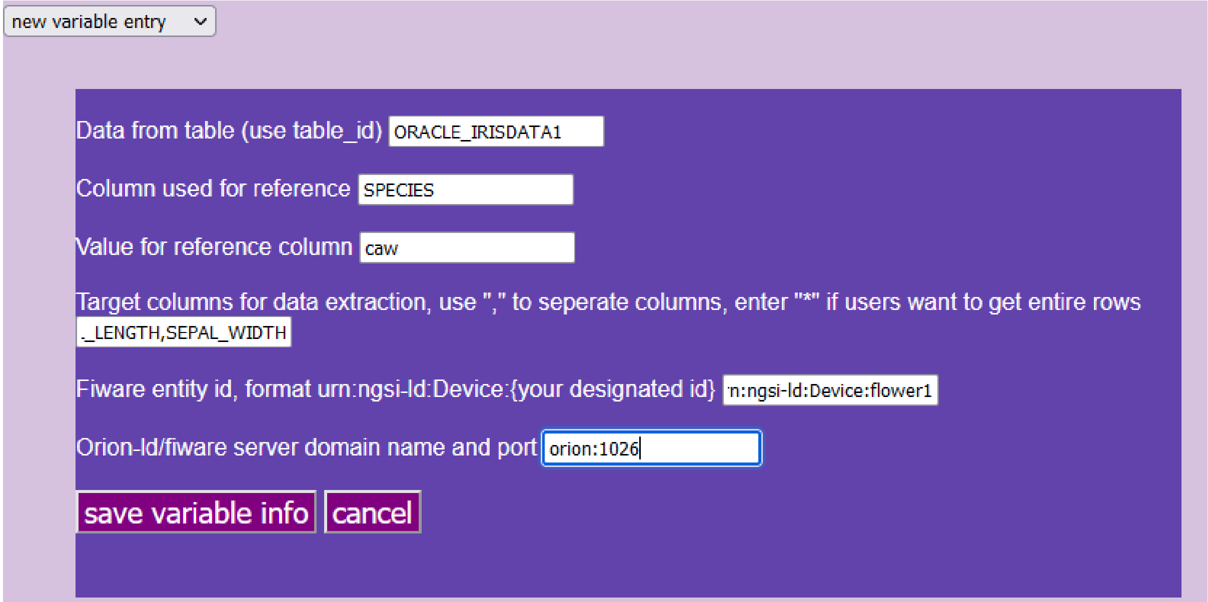
Target columns for data extraction, use "," to seperate columns, enter "*" if users want to get entire rows

Fiware entity id, format urn:ngsi-ld:Device:{your designated id}

Orion-ld/fiware server domain name and port

The table_id must be the table_id of one of defined database table (either legacy or mongodb table).

For example, defining a new entity with id “urn:ngsi-ld:Device:flower1” that retrieves the SEPAL_LENGTH, SEPAL_WIDTH of SPECIES with name “caw” from table ORACLE_IRISDATA1



Data from table (use table_id) ORACLE_IRISDATA1

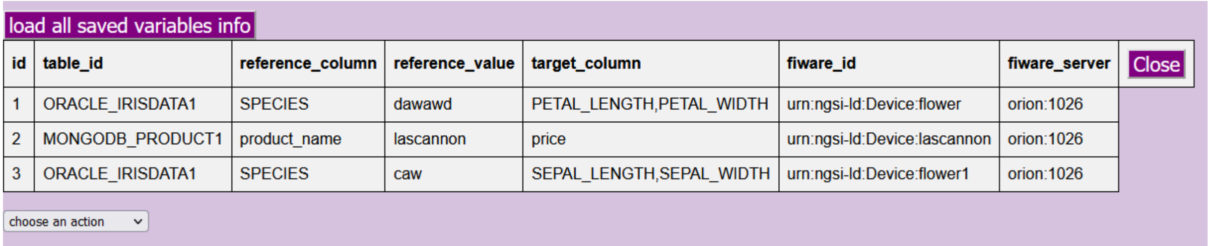
Column used for reference SPECIES

Value for reference column caw

Target columns for data extraction, use "," to seperate columns, enter "*" if users want to get entire rows
_LENGTH,SEPAL_WIDTH

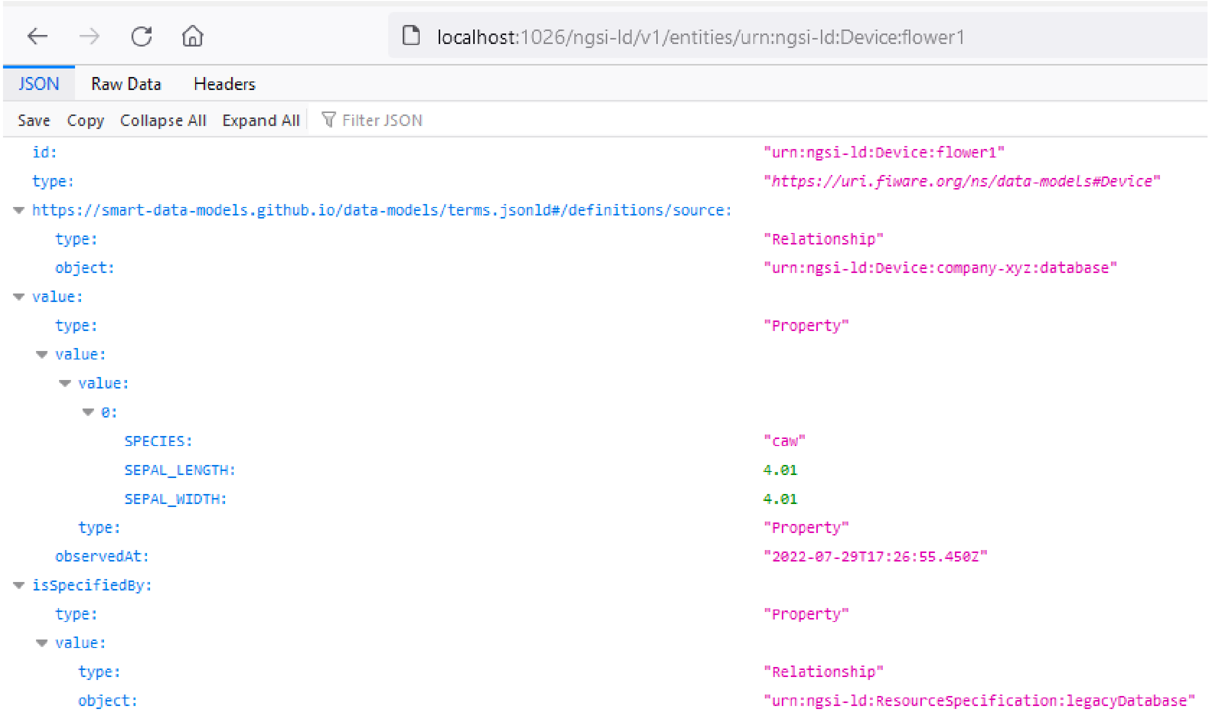
Fiware entity id, format urn:ngsi-ld:Device:{your designated id} n:ngsi-ld:Device:flower1

Orion-ld/fiware server domain name and port orion:1026



id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	orion:1026
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	orion:1026
3	ORACLE_IRISDATA1	SPECIES	caw	SEPAL_LENGTH,SEPAL_WIDTH	urn:ngsi-ld:Device:flower1	orion:1026

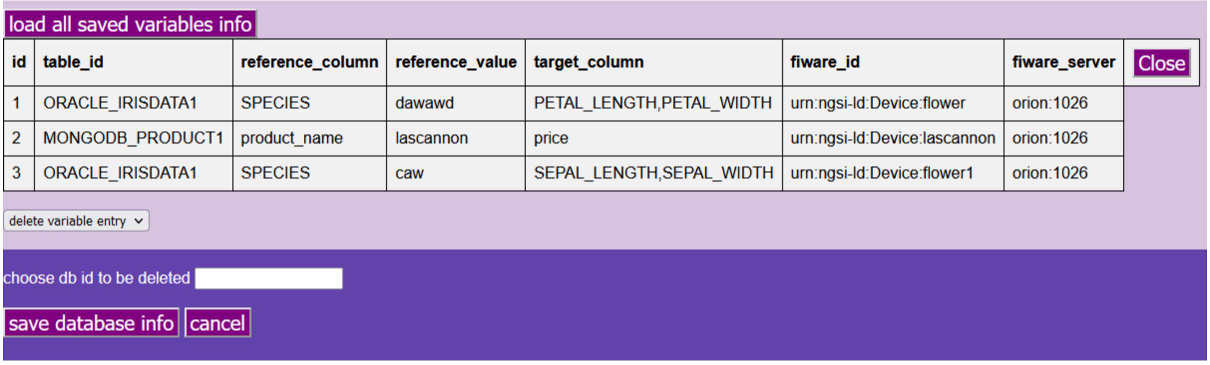
Result from orion server


localhost:1026/ngsi-ld/v1/entities/urn:ngsi-ld:Device:flower1
JSON
Raw Data
Headers
Save
Copy
Collapse All
Expand All
Filter JSON

```
id:                                                                     "urn:ngsi-ld:Device:flower1"
type:                                                                   "https://uri.fiware.org/ns/data-models#Device"
▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/source:
    type:                                                               "Relationship"
    object:                                                             "urn:ngsi-ld:Device:company-xyz:database"
▼ value:
    type:                                                               "Property"
  ▼ value:
    ▼ value:
      ▼ 0:
          SPECIES:                                                      "caw"
          SEPAL_LENGTH:                                                 4.01
          SEPAL_WIDTH:                                                  4.01
      type:                                                             "Property"
    observedAt:                                                         "2022-07-29T17:26:55.450Z"
▼ isSpecifiedBy:
    type:                                                               "Property"
  ▼ value:
      type:                                                             "Relationship"
      object:                                                           "urn:ngsi-ld:ResourceSpecification:legacyDatabase"
```

Delete entry

To delete entry, choose “delete variable entry” and provide the id of the entry that need deleting



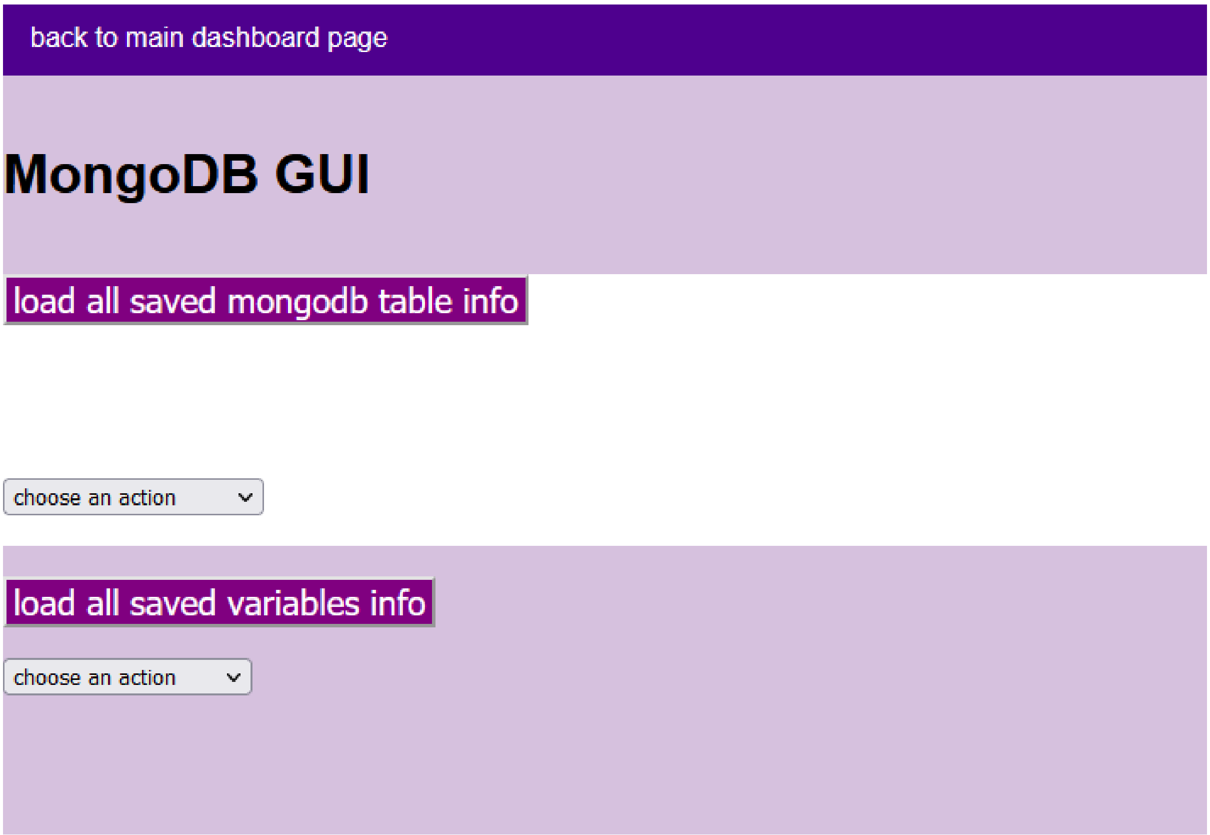
id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	orion:1026
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	orion:1026
3	ORACLE_IRISDATA1	SPECIES	caw	SEPAL_LENGTH,SEPAL_WIDTH	urn:ngsi-ld:Device:flower1	orion:1026

choose db id to be deleted

MongoDB database interface

MongoDB database interface serves two purposes:

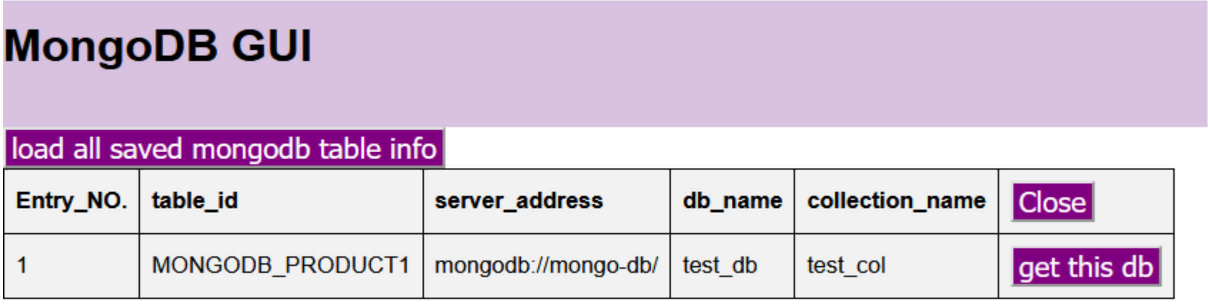
- Define database entry with parameters for saving the data within DCF (only apply for mongoDB)
- Define fiware entities for data extraction from saved database (this functionality is the same as the one in legacy interface)


back to main dashboard page
MongoDB GUI
load all saved mongodb table info
choose an action
load all saved variables info
choose an action

Adding new mongoDB info entry for saving data or deleting existing entry

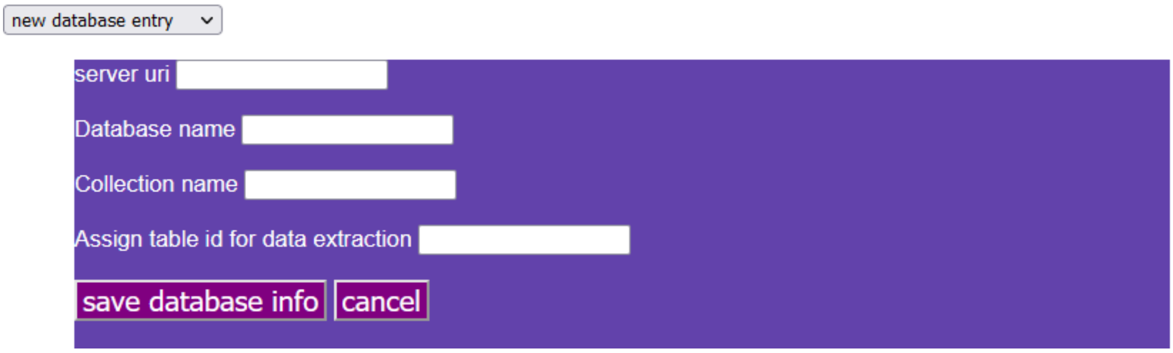
Adding new entry

To add new entry, select “new database entry”



MongoDB GUI

Entry_NO.	table_id	server_address	db_name	collection_name	Close
1	MONGODB_PRODUCT1	mongodb://mongo-db/	test_db	test_col	get this db



server uri

Database name

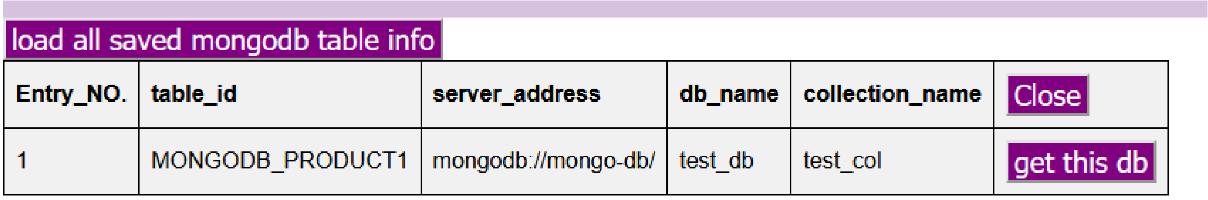
Collection name

Assign table id for data extraction

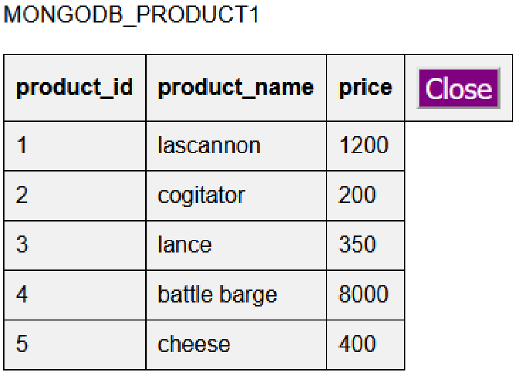
Users need to provide needed parameters and unique table id, the table id is any of users' choosing and must be unique.

After the entry is defined and data is retrieved, the data can be viewed and used for data extraction.

For example, the content of table_id MONGODB_PRODUCT1



Entry_NO.	table_id	server_address	db_name	collection_name	Close
1	MONGODB_PRODUCT1	mongodb://mongo-db/	test_db	test_col	get this db

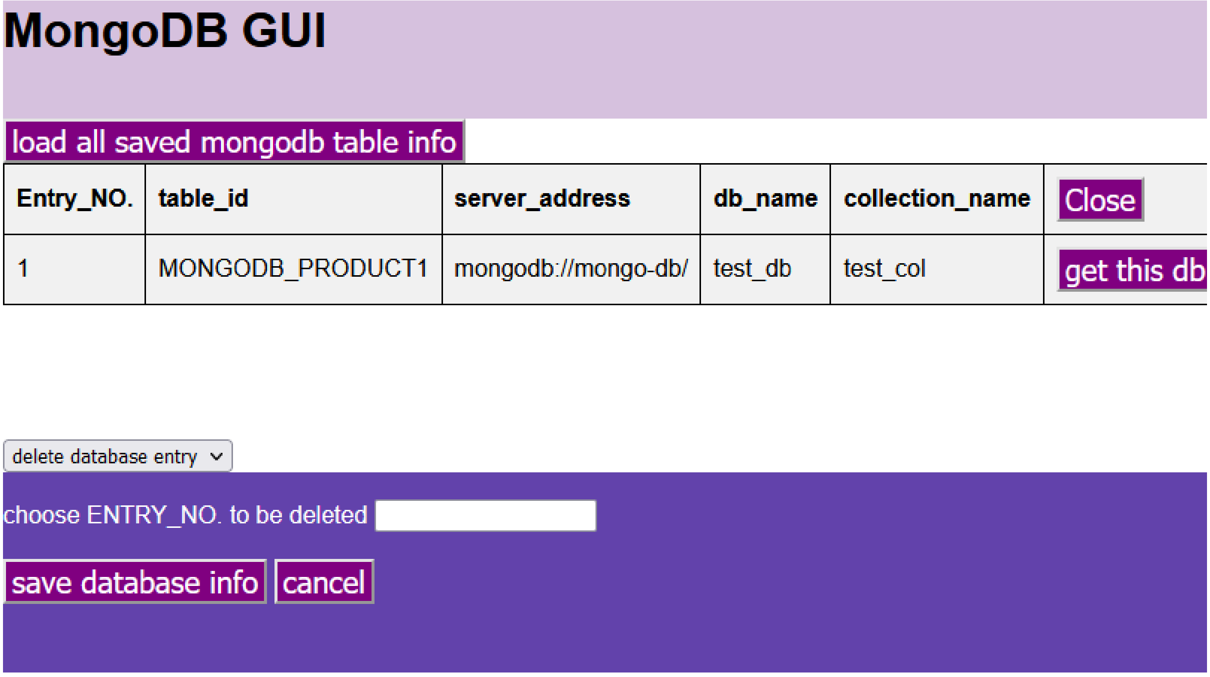


MONGODB_PRODUCT1

product_id	product_name	price	Close
1	lascannon	1200	
2	cogitator	200	
3	lance	350	
4	battle barge	8000	
5	cheese	400	

Deleting existing entry

To delete existing entry, select “delete database entry” and provide the ENTRY_NO. of the entry that needs deleting



MongoDB GUI

Entry_NO.	table_id	server_address	db_name	collection_name	Close
1	MONGODB_PRODUCT1	mongodb://mongo-db/	test_db	test_col	get this db

delete database entry

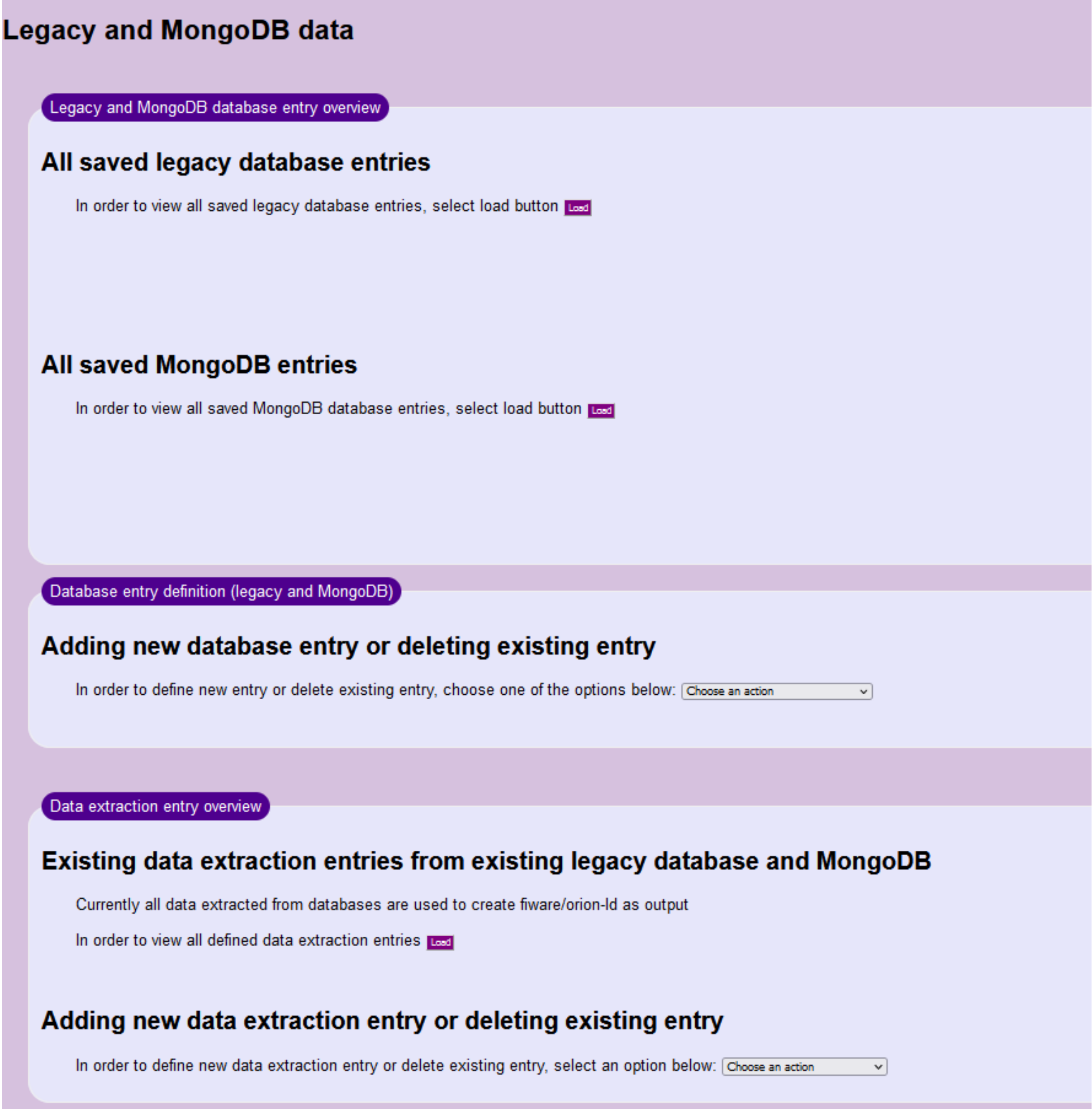
choose ENTRY_NO. to be deleted

Adding new parameters entry for data retrieval or deleting existing entry

The procedures are the same as in legacy interface

Update note on Legacy and Mongo DB database interface

Since version 1.6, both Legacy and Mongo DB interfaces are merged into the same interface.



Legacy and MongoDB data

Legacy and MongoDB database entry overview

All saved legacy database entries

In order to view all saved legacy database entries, select load button

All saved MongoDB entries

In order to view all saved MongoDB database entries, select load button

Database entry definition (legacy and MongoDB)

Adding new database entry or deleting existing entry

In order to define new entry or delete existing entry, choose one of the options below: Choose an action

Data extraction entry overview

Existing data extraction entries from existing legacy database and MongoDB

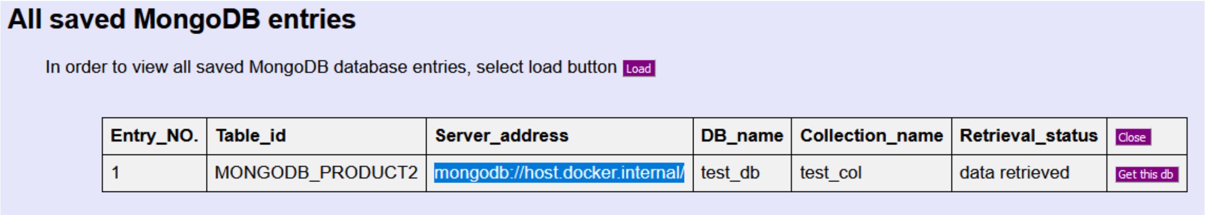
Currently all data extracted from databases are used to create fiware/orion-ld as output

In order to view all defined data extraction entries

Adding new data extraction entry or deleting existing entry

In order to define new data extraction entry or delete existing entry, select an option below: Choose an action

While procedures remain the same for all entries, there is a change to MongoDB domain for data extraction. Due to DCF container is no longer placed within the same network as dockerized Mongo-DB and Fiware orion-ld container, the domain for MongoDB is “mongodb://host.docker.internal/” and fiware orion-ld domain “host.docker.internal:1026”

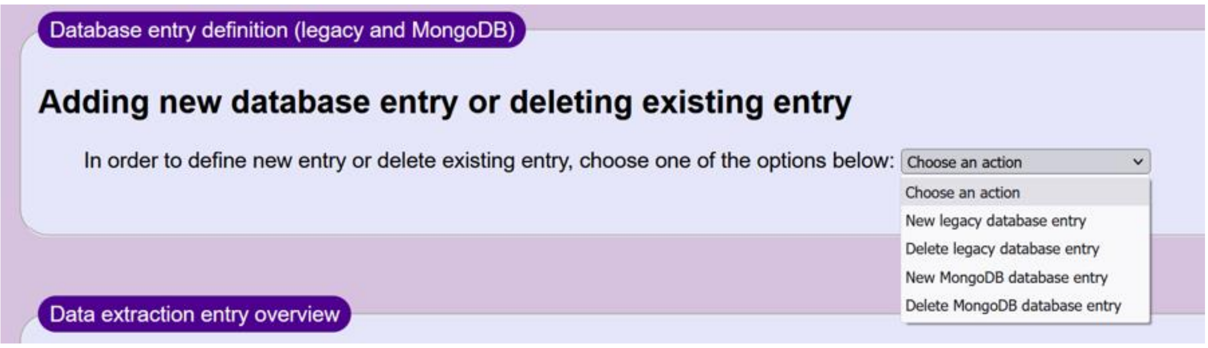


All saved MongoDB entries

In order to view all saved MongoDB database entries, select load button

Entry_NO.	Table_id	Server_address	DB_name	Collection_name	Retrieval_status	Close
1	MONGODB_PRODUCT2	mongodb://host.docker.internal/	test_db	test_col	data retrieved	Get this db

Menu selection for database entries manipulation



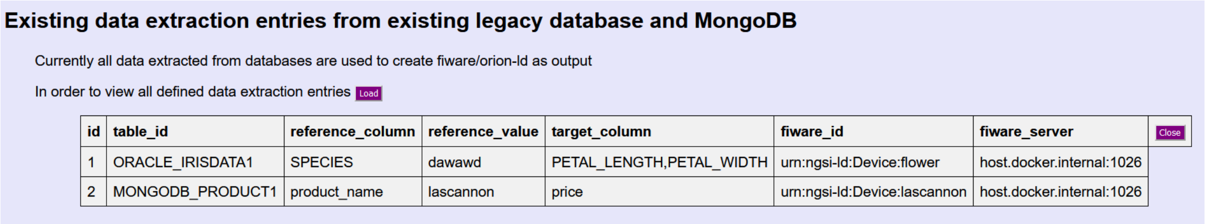
Database entry definition (legacy and MongoDB)

Adding new database entry or deleting existing entry

In order to define new entry or delete existing entry, choose one of the options below:

Choose an action
Choose an action
New legacy database entry
Delete legacy database entry
New MongoDB database entry
Delete MongoDB database entry
Data extraction entry overview

Domain name for entity publishing.



Existing data extraction entries from existing legacy database and MongoDB

Currently all data extracted from databases are used to create fiware/orion-ld as output

In order to view all defined data extraction entries Load

id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server	Close
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	host.docker.internal:1026	
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	host.docker.internal:1026	