DCF

Data Collection Framework

User Manual

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.

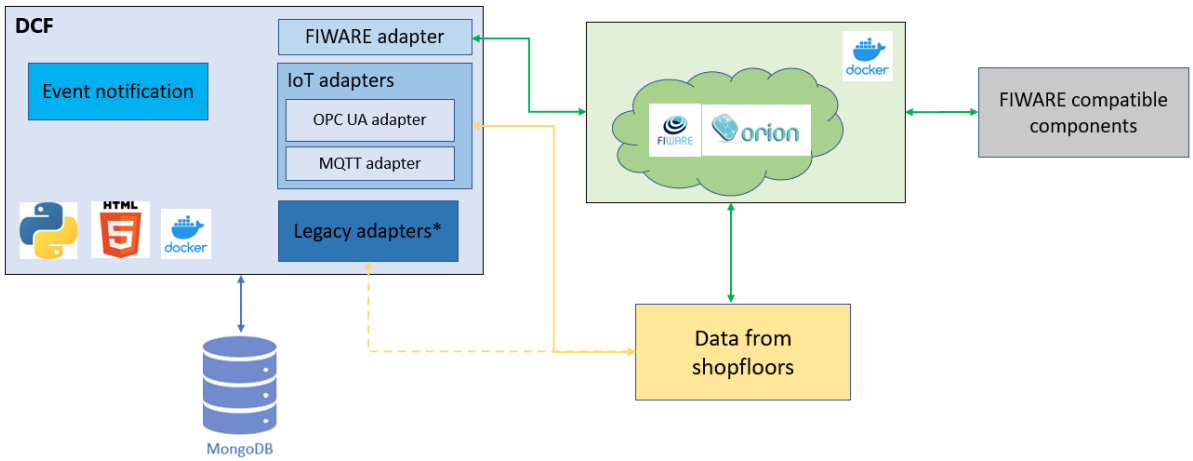


Figure 1: DCF architecture

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.

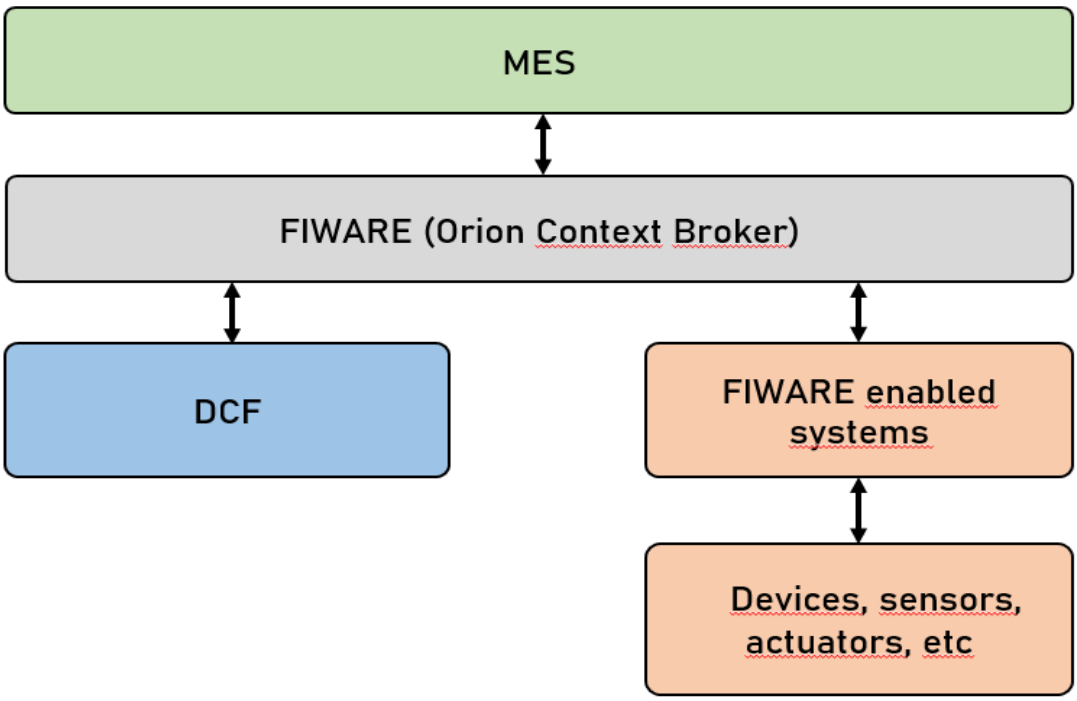
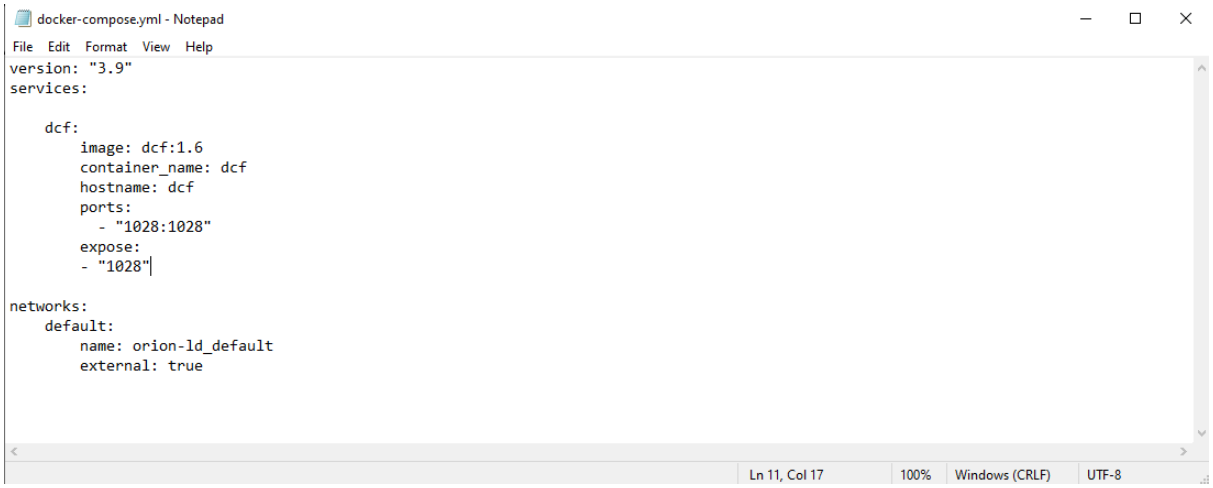


Figure 2: DCF communication with other components

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.

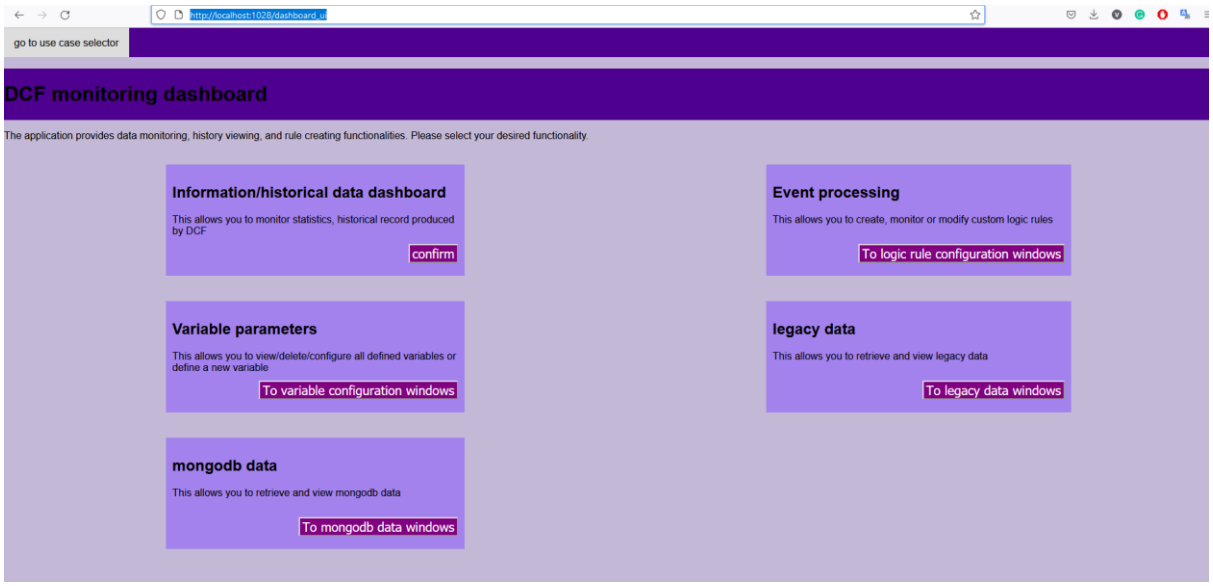
Before composing the image, the networks parameter in docker-compose.yml file needs to be configurated. If users want use containerized MongoDB and Fiware-Orion server images without knowing their IP address, the DCF container needs to be put into the same existing docker network as MongoDB and Fiware-Orion by replacing networks/default/name parameter with name of the network name, with this, the domain name of the images can be used as domain address without the need of exact IP addresses or web addresses. If users use noncontainerized Fiware/Orion-ld and MongoDB server image or do not want to add the DCF image to any network (only use sperate network), the entire networks section needs to be deleted.



USER INTERFACE

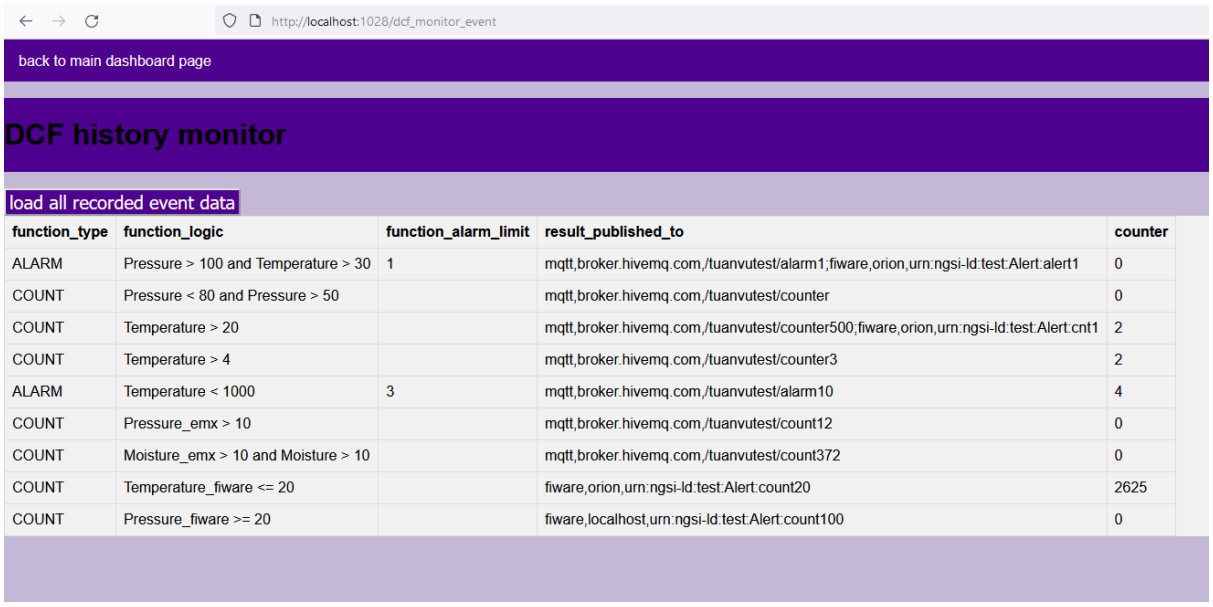
DCF also provides a GUI where users can monitor events and define new operations.

First, the dashboard is used as navigation panel to different functionalities of DCF



Information/historical data dashboard

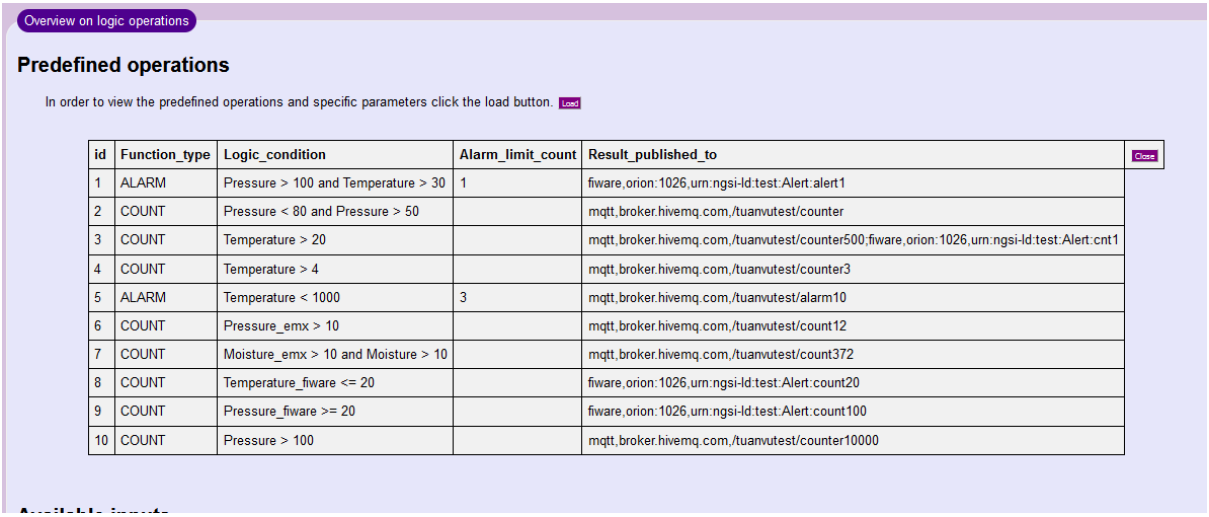
This dashboard allows users to monitor output counters of defined logic operations and their parameters such as logic conditions and output publish endpoint.



function_type	function_logic	function_alarm_limit	result_published_to	counter
ALARM	Pressure > 100 and Temperature > 30	1	mqtt,broker.hivemq.com,/tuanvutest/alarm1;fiware,orion,urn:ngsi-ld:test:Alert:alert1	0
COUNT	Pressure < 80 and Pressure > 50		mqtt,broker.hivemq.com,/tuanvutest/counter	0
COUNT	Temperature > 20		mqtt,broker.hivemq.com,/tuanvutest/counter500;fiware,orion,urn:ngsi-ld:test:Alert:cnt1	2
COUNT	Temperature > 4		mqtt,broker.hivemq.com,/tuanvutest/counter3	2
ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10	4
COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12	0
COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372	0
COUNT	Temperature_fiware <= 20		fiware,orion,urn:ngsi-ld:test:Alert:count20	2625
COUNT	Pressure_fiware >= 20		fiware,localhost,urn:ngsi-ld:test:Alert:count100	0

Event processing interface

This interface allows users to define new logic operations or modify existing operations.



Overview on logic operations

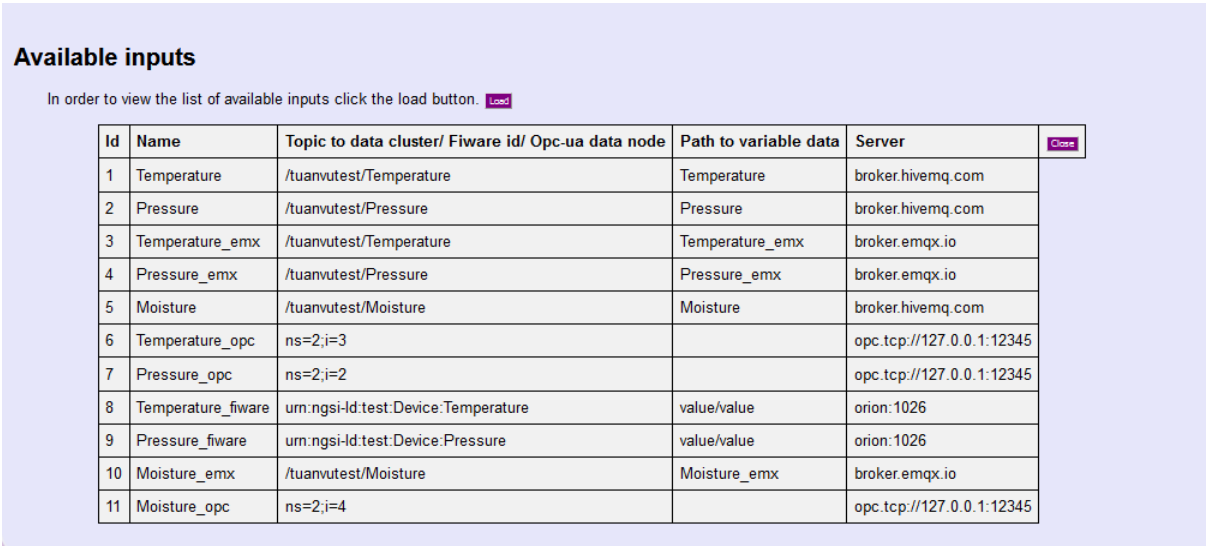
Predefined operations

In order to view the predefined operations and specific parameters click the load button. Load

id	Function_type	Logic_condition	Alarm_limit_count	Result_published_to
1	ALARM	Pressure > 100 and Temperature > 30	1	fiware,orion:1026,urn:ngsi-ld:test:Alert:alert1
2	COUNT	Pressure < 80 and Pressure > 50		mqtt,broker.hivemq.com,/tuanvutest/counter
3	COUNT	Temperature > 20		mqtt,broker.hivemq.com,/tuanvutest/counter500;fiware,orion:1026,urn:ngsi-ld:test:Alert:cnt1
4	COUNT	Temperature > 4		mqtt,broker.hivemq.com,/tuanvutest/counter3
5	ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10
6	COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12
7	COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372
8	COUNT	Temperature_fiware <= 20		fiware,orion:1026,urn:ngsi-ld:test:Alert:count20
9	COUNT	Pressure_fiware >= 20		fiware,orion:1026,urn:ngsi-ld:test:Alert:count100
10	COUNT	Pressure > 100		mqtt,broker.hivemq.com,/tuanvutest/counter10000

Close

To assist users in these tasks, list of defined variables is also provided in the interface.



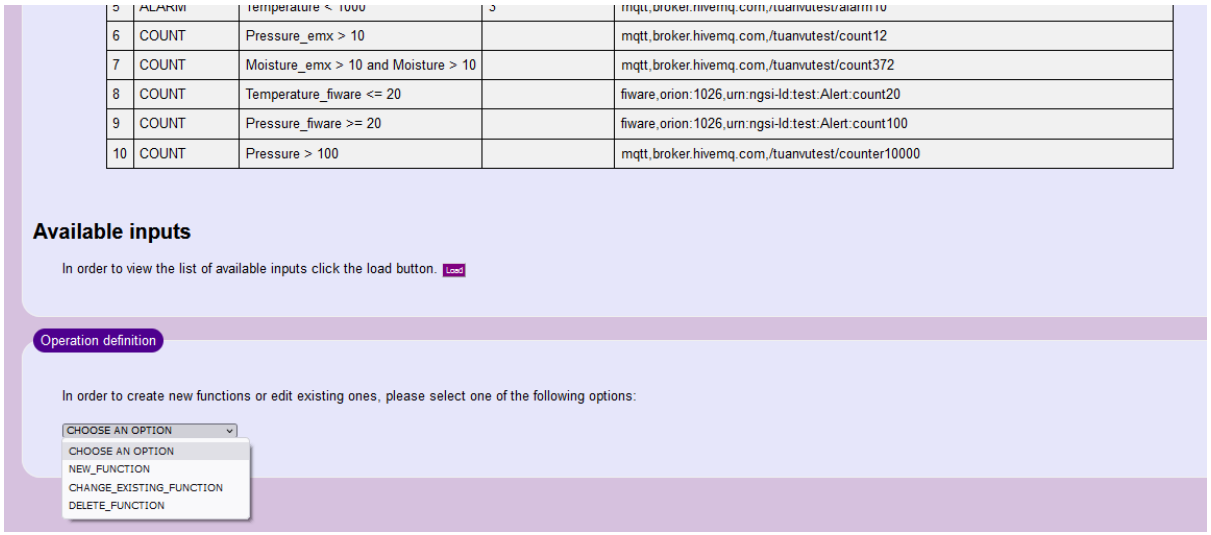
Available inputs

In order to view the list of available inputs click the load button. Load

Id	Name	Topic to data cluster/ Fiware id/ Opc-ua data node	Path to variable data	Server
1	Temperature	/tuanvutest/Temperature	Temperature	broker.hivemq.com
2	Pressure	/tuanvutest/Pressure	Pressure	broker.hivemq.com
3	Temperature_emx	/tuanvutest/Temperature	Temperature_emx	broker.emqx.io
4	Pressure_emx	/tuanvutest/Pressure	Pressure_emx	broker.emqx.io
5	Moisture	/tuanvutest/Moisture	Moisture	broker.hivemq.com
6	Temperature_opc	ns=2;i=3		opc.tcp://127.0.0.1:12345
7	Pressure_opc	ns=2;i=2		opc.tcp://127.0.0.1:12345
8	Temperature_fiware	urn:ngsi-ld:test:Device:Temperature	value/value	orion:1026
9	Pressure_fiware	urn:ngsi-ld:test:Device:Pressure	value/value	orion:1026
10	Moisture_emx	/tuanvutest/Moisture	Moisture_emx	broker.emqx.io
11	Moisture_opc	ns=2;i=4		opc.tcp://127.0.0.1:12345

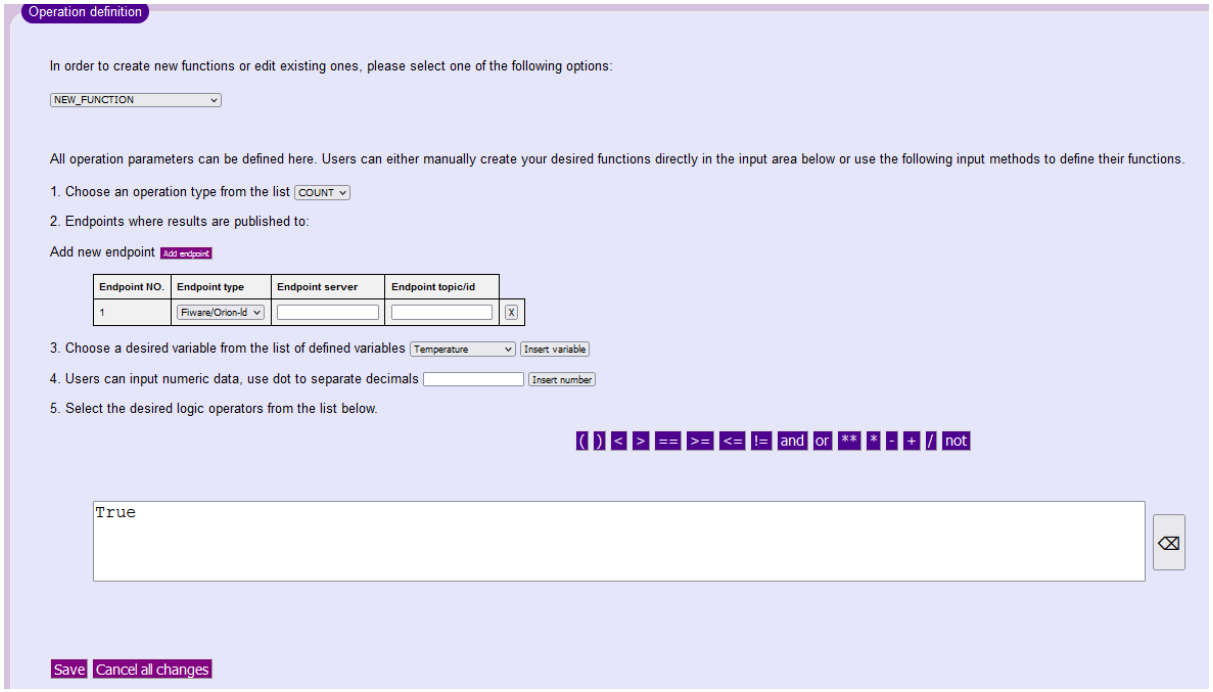
Close

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



5	ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10
6	COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12
7	COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372
8	COUNT	Temperature_fiware <= 20		fiware,orion:1026,urn:ngsi-ld:test:Alert:count20
9	COUNT	Pressure_fiware >= 20		fiware,orion:1026,urn:ngsi-ld:test:Alert:count100
10	COUNT	Pressure > 100		mqtt,broker.hivemq.com,/tuanvutest/counter10000

Creating new operation



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

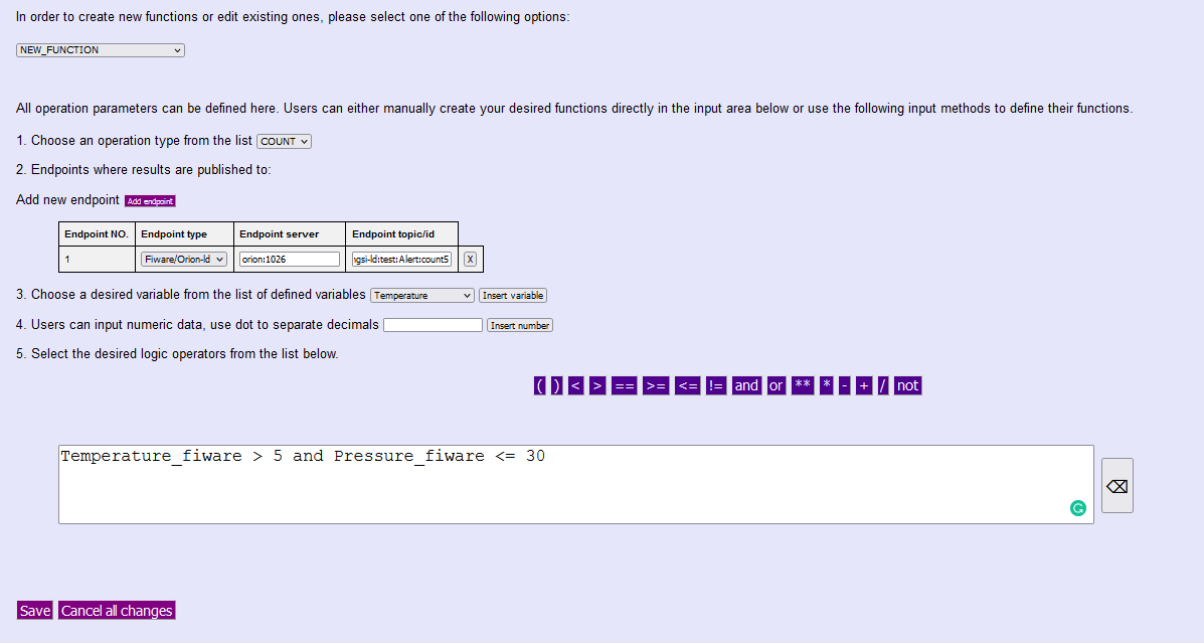
Section 5 is the place where logic arguments are created, the default condition for new operation is always “True”.

There are 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

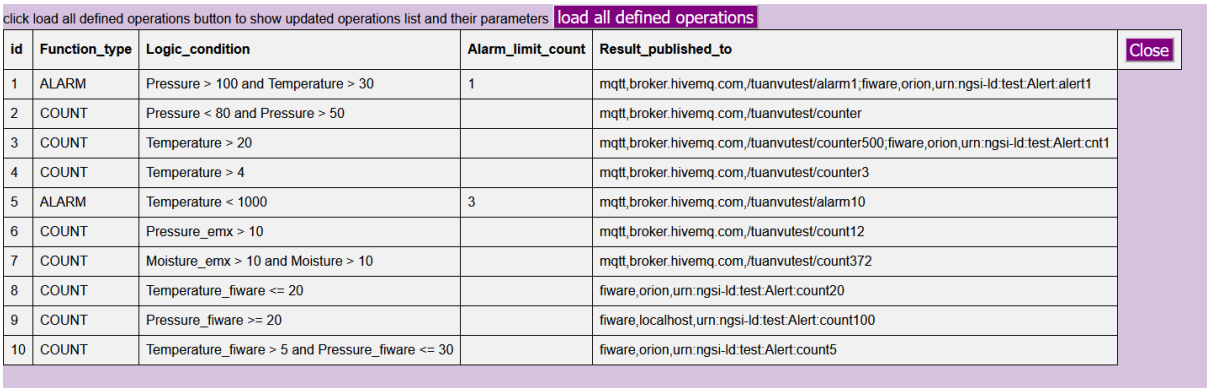
logic operators from the logic operator list. It is highly recommended to use these features to ensure the data format is followed so the program can parse the data. To finalize, the “confirm logic change” button is also needed to be chosen.

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 either “fiware” or “mqtt”. In this case, the result needs to be publish to fiware with the entity id “urn:ngsi-ld:test:Alert:count5”, on my containerized fiware/orion-ld server so the server should be orion:1026

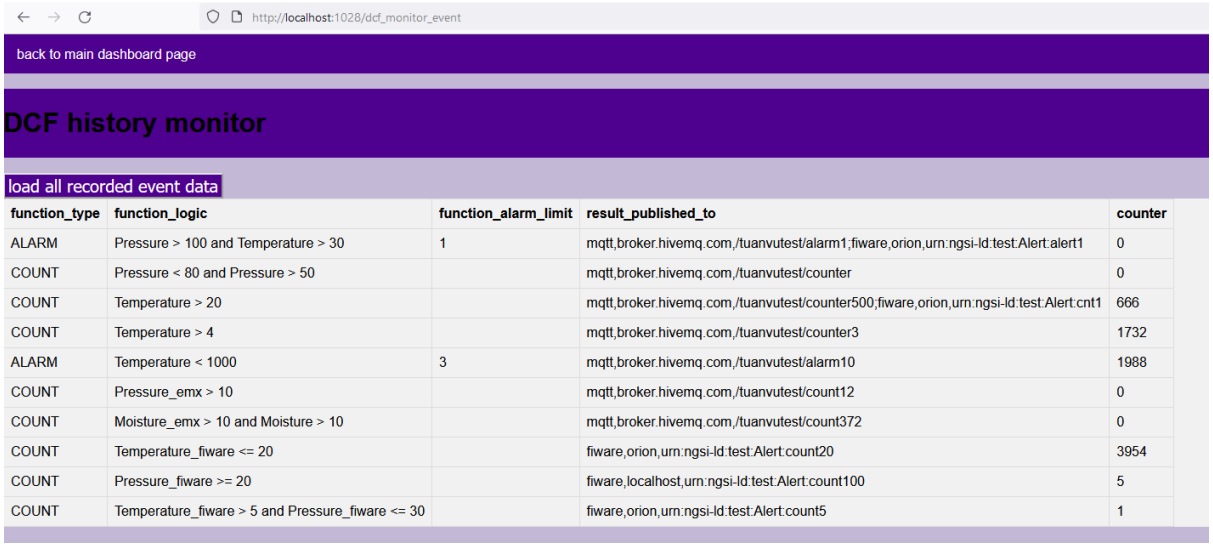


Push “add newly defined function” button to update the data.

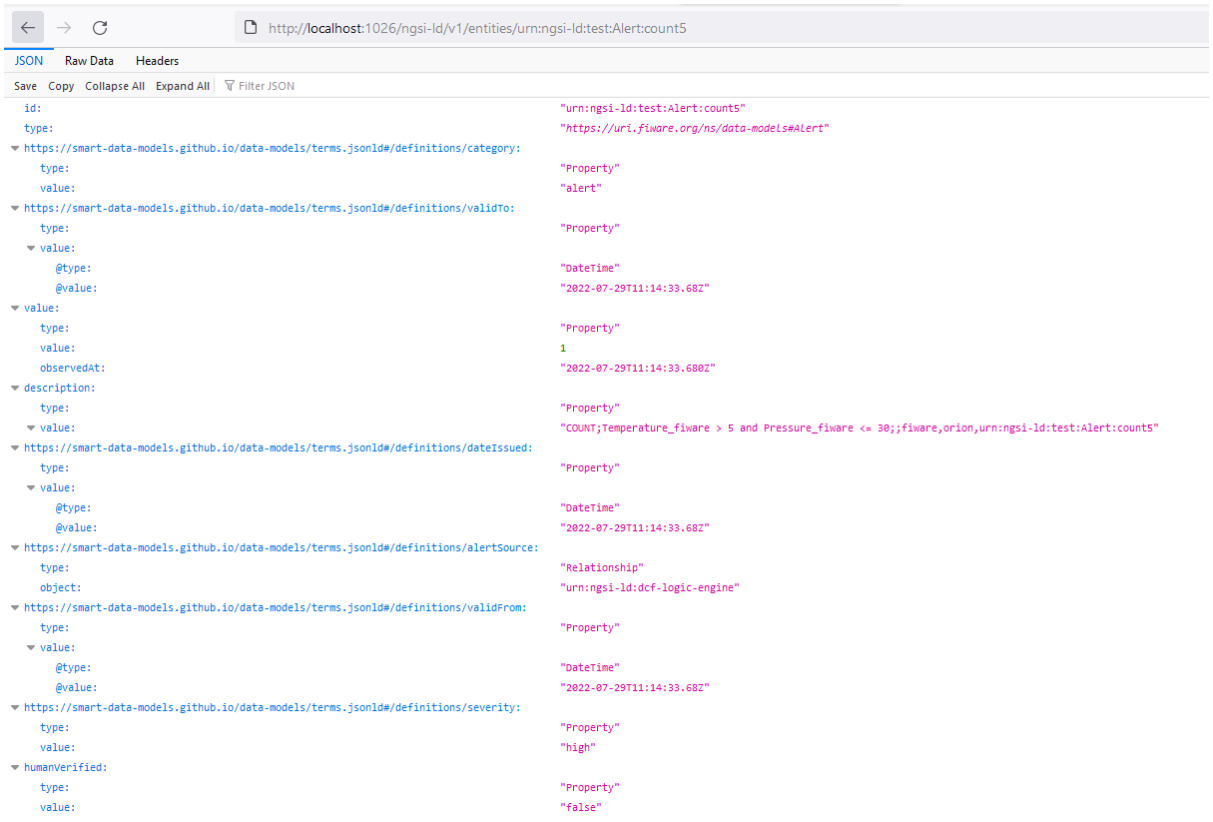


id	Function_type	Logic_condition	Alarm_limit_count	Result_published_to
1	ALARM	Pressure > 100 and Temperature > 30	1	mqtt,broker.hivemq.com,/tuanvutest/alarm1;fiware,orion,urn:ngsi-ld:test:Alert:alert1
2	COUNT	Pressure < 80 and Pressure > 50		mqtt,broker.hivemq.com,/tuanvutest/counter
3	COUNT	Temperature > 20		mqtt,broker.hivemq.com,/tuanvutest/counter500;fiware,orion,urn:ngsi-ld:test:Alert:cnt1
4	COUNT	Temperature > 4		mqtt,broker.hivemq.com,/tuanvutest/counter3
5	ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10
6	COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12
7	COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372
8	COUNT	Temperature_fiware <= 20		fiware,orion,urn:ngsi-ld:test:Alert:count20
9	COUNT	Pressure_fiware >= 20		fiware,localhost,urn:ngsi-ld:test:Alert:count100
10	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30		fiware,orion,urn:ngsi-ld:test:Alert:count5

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



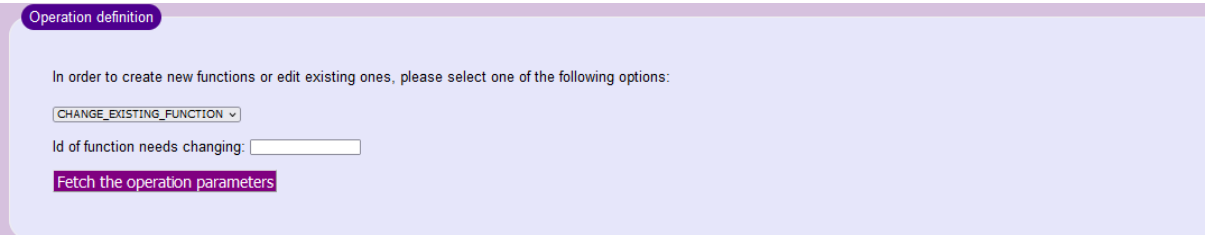
function_type	function_logic	function_alarm_limit	result_published_to	counter
ALARM	Pressure > 100 and Temperature > 30	1	mqtt,broker.hivemq.com,/tuanvutest/alarm1;fiware,orion,urn:ngsi-ld:test:Alert:alert1	0
COUNT	Pressure < 80 and Pressure > 50		mqtt,broker.hivemq.com,/tuanvutest/counter	0
COUNT	Temperature > 20		mqtt,broker.hivemq.com,/tuanvutest/counter500;fiware,orion,urn:ngsi-ld:test:Alert:cnt1	666
COUNT	Temperature > 4		mqtt,broker.hivemq.com,/tuanvutest/counter3	1732
ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10	1988
COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12	0
COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372	0
COUNT	Temperature_fiware <= 20		fiware,orion,urn:ngsi-ld:test:Alert:count20	3954
COUNT	Pressure_fiware >= 20		fiware,localhost,urn:ngsi-ld:test:Alert:count100	5
COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30		fiware,orion,urn:ngsi-ld:test:Alert:count5	1



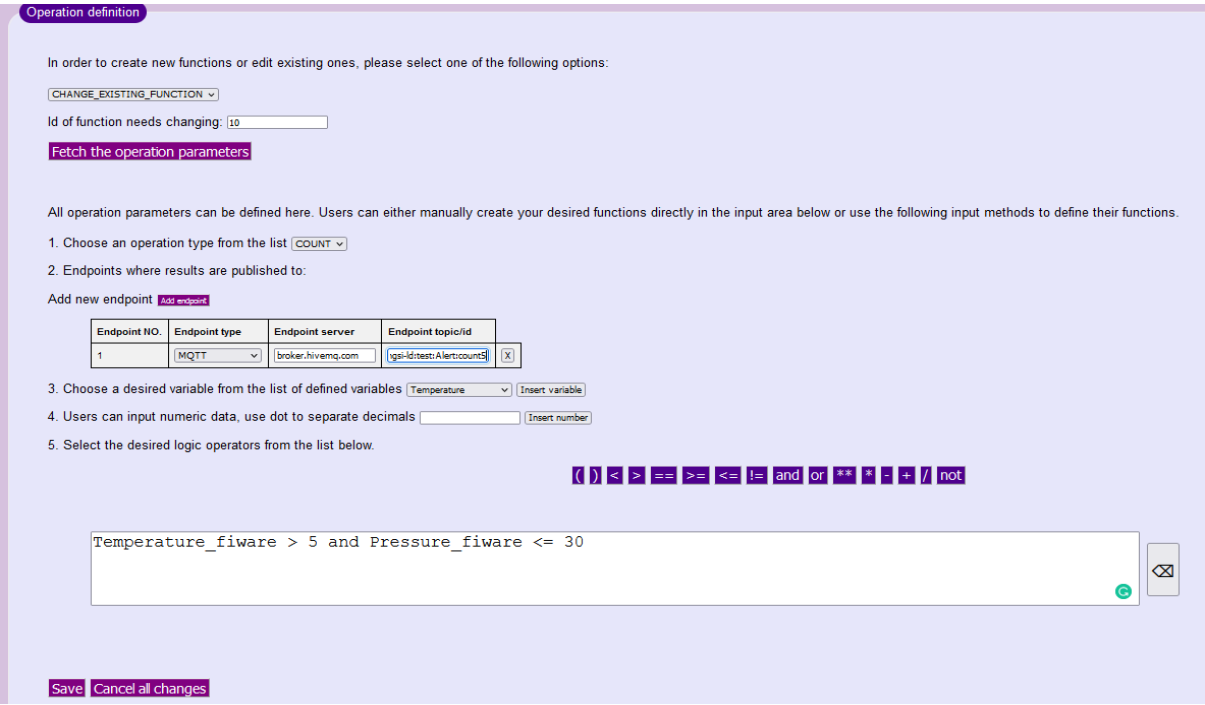
The operation is registered and output is published.

Modifying existing operation

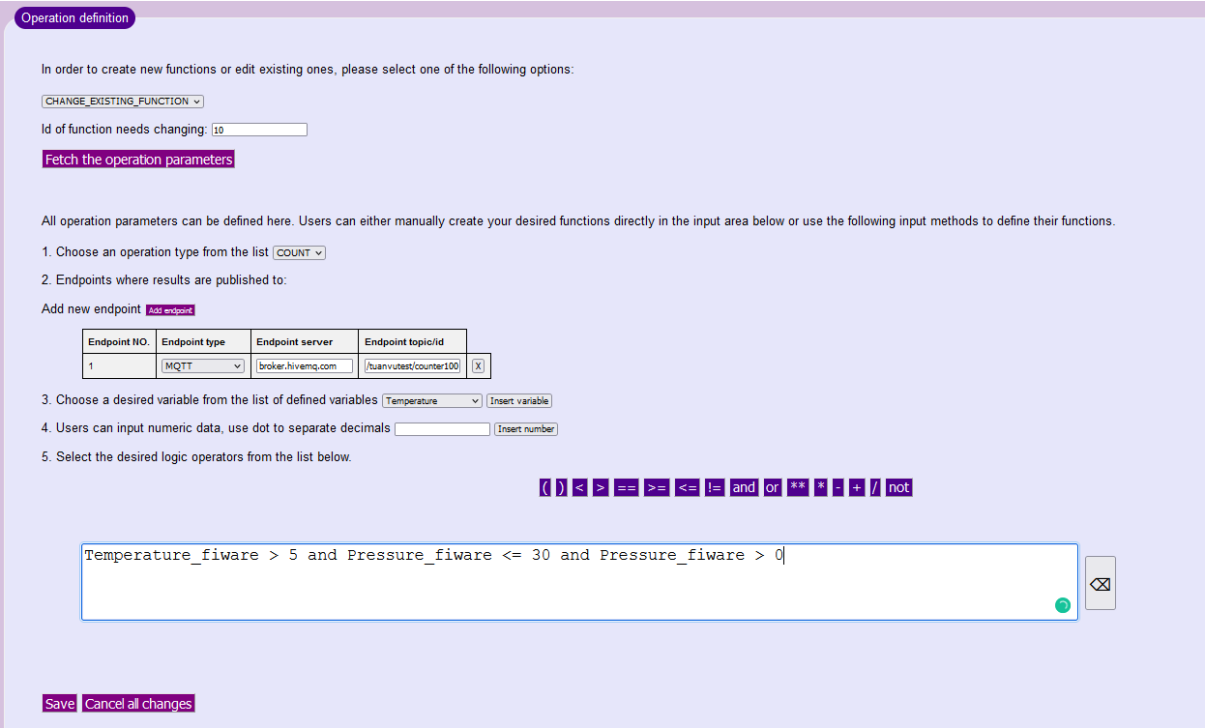
To modifying existing operation, select “CHANGE_EXISTING_FUNCTION” option



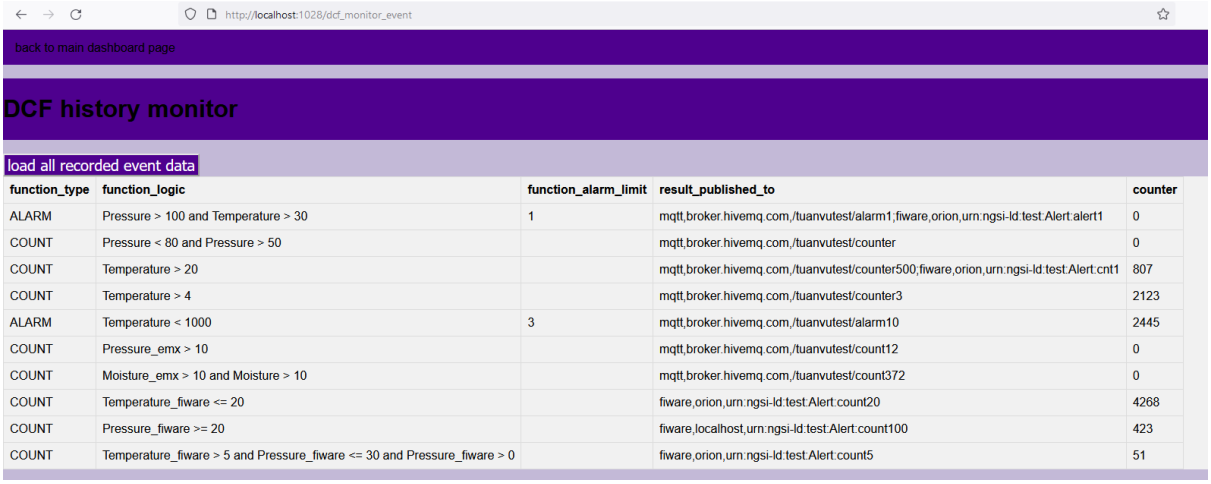
From here, users need to provide the id of the operation that needs to be modified, for example, the operation last created with id 10 needs to add new logic condition, after providing the id, “get the function” button need to be clicked to retrieve the operation parameter



New condition is added



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.

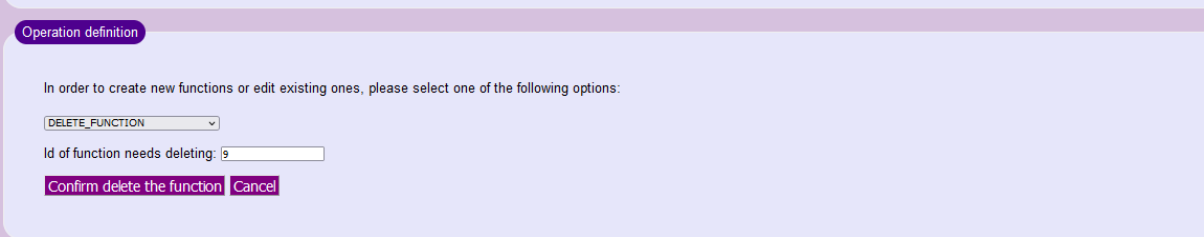


function_type	function_logic	function_alarm_limit	result_published_to	counter
ALARM	Pressure > 100 and Temperature > 30	1	mqtt,broker.hivemq.com,/tuanvutest/alarm1;fiware,orion,urn:ngsi-ld:test:Alert:alert1	0
COUNT	Pressure < 80 and Pressure > 50		mqtt,broker.hivemq.com,/tuanvutest/counter	0
COUNT	Temperature > 20		mqtt,broker.hivemq.com,/tuanvutest/counter500;fiware,orion,urn:ngsi-ld:test:Alert:cnt1	807
COUNT	Temperature > 4		mqtt,broker.hivemq.com,/tuanvutest/counter3	2123
ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10	2445
COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12	0
COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372	0
COUNT	Temperature_fiware <= 20		fiware,orion,urn:ngsi-ld:test:Alert:count20	4268
COUNT	Pressure_fiware >= 20		fiware,localhost,urn:ngsi-ld:test:Alert:count100	423
COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30 and Pressure_fiware > 0		fiware,orion,urn:ngsi-ld:test:Alert:count5	51

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 9 with logic condition “Pressure_fiware >= 20”



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

Predefined operations

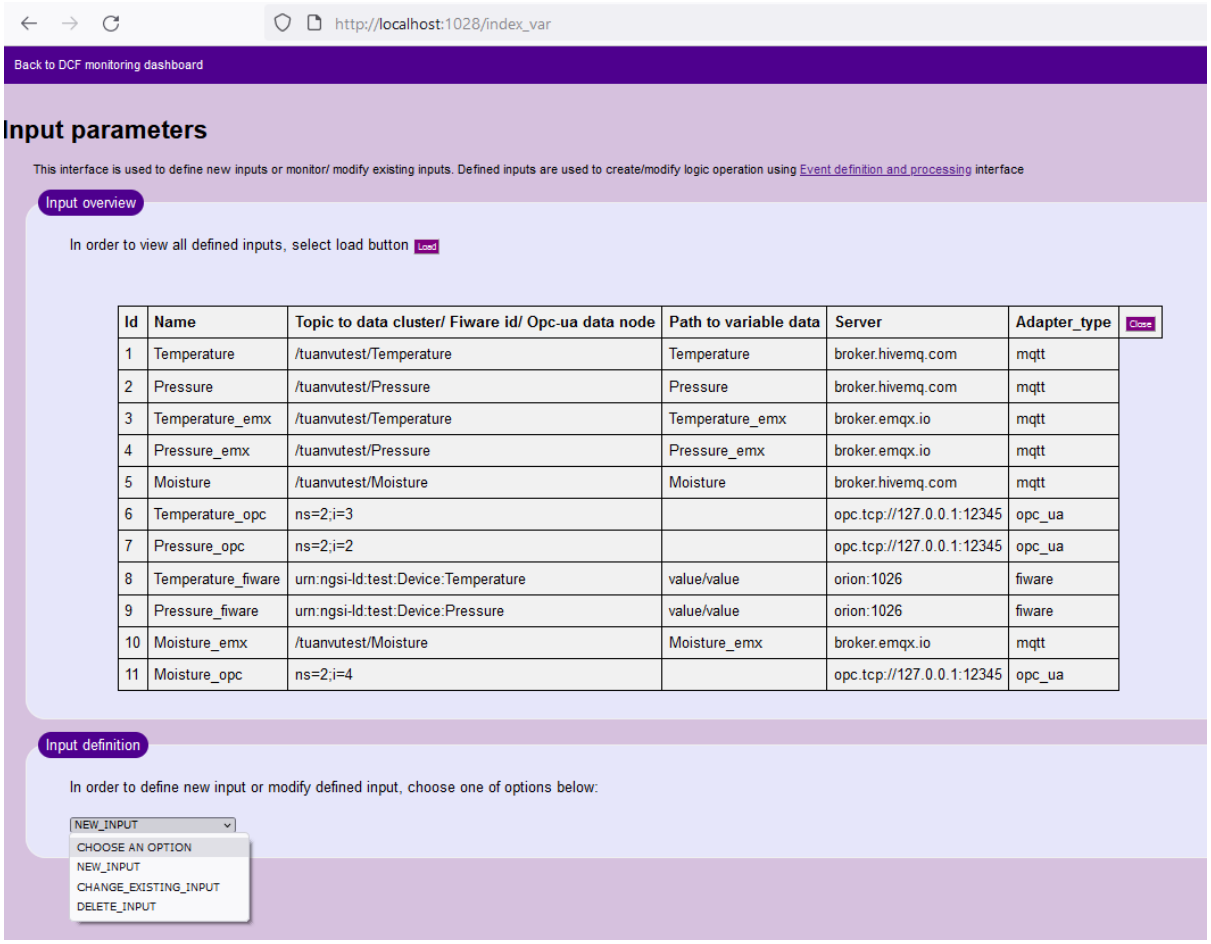
In order to view the predefined operations and specific parameters click the load button. Load

id	Function_type	Logic_condition	Alarm_limit_count	Result_published_to	Close
1	ALARM	Pressure > 100 and Temperature > 30	1	fiware,orion:1026,urn:ngsi-ld:test:Alert:alert1	
2	COUNT	Pressure < 80 and Pressure > 50		mqtt,broker.hivemq.com,/tuanvutest/counter	
3	COUNT	Temperature > 20		mqtt,broker.hivemq.com,/tuanvutest/counter500;fiware,orion:1026,urn:ngsi-ld:test:Alert:cnt1	
4	COUNT	Temperature > 4		mqtt,broker.hivemq.com,/tuanvutest/counter3	
5	ALARM	Temperature < 1000	3	mqtt,broker.hivemq.com,/tuanvutest/alarm10	
6	COUNT	Pressure_emx > 10		mqtt,broker.hivemq.com,/tuanvutest/count12	
7	COUNT	Moisture_emx > 10 and Moisture > 10		mqtt,broker.hivemq.com,/tuanvutest/count372	
8	COUNT	Temperature_fiware <= 20		fiware,orion:1026,urn:ngsi-ld:test:Alert:count20	
9	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30 and Pressure_fiware > 0		mqtt,broker.hivemq.com,urn:ngsi-ld:test:Alert:count5	

Available inputs

Input parameters interface

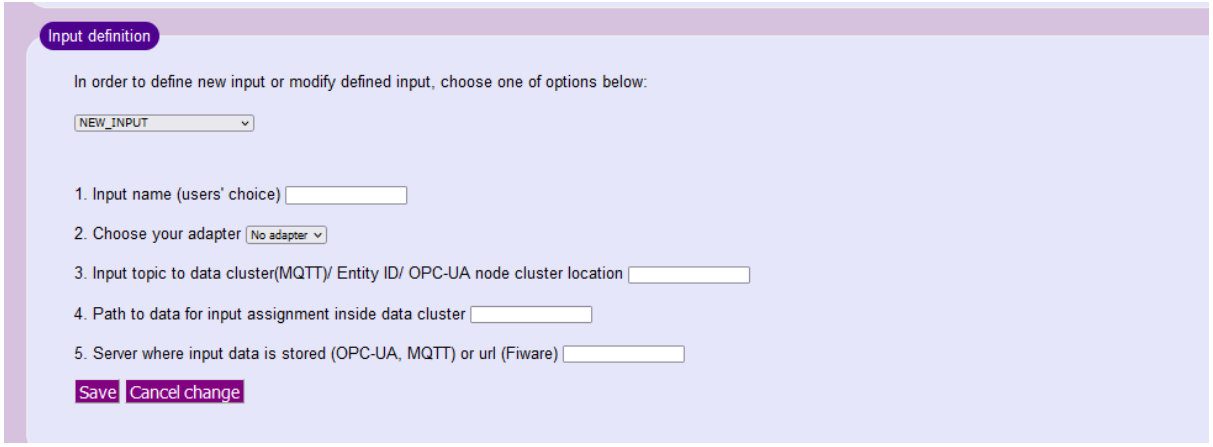
Input parameters interface is used to define, delete or modify the input variables that are used for logic operations



Id	Name	Topic to data cluster/ Fiware id/ Opc-ua data node	Path to variable data	Server	Adapter_type
1	Temperature	/tuanvutest/Temperature	Temperature	broker.hivemq.com	mqtt
2	Pressure	/tuanvutest/Pressure	Pressure	broker.hivemq.com	mqtt
3	Temperature_emx	/tuanvutest/Temperature	Temperature_emx	broker.emqx.io	mqtt
4	Pressure_emx	/tuanvutest/Pressure	Pressure_emx	broker.emqx.io	mqtt
5	Moisture	/tuanvutest/Moisture	Moisture	broker.hivemq.com	mqtt
6	Temperature_opc	ns=2;i=3		opc.tcp://127.0.0.1:12345	opc_ua
7	Pressure_opc	ns=2;i=2		opc.tcp://127.0.0.1:12345	opc_ua
8	Temperature_fiware	urn:ngsi-ld:test:Device:Temperature	value/value	orion:1026	fiware
9	Pressure_fiware	urn:ngsi-ld:test:Device:Pressure	value/value	orion:1026	fiware
10	Moisture_emx	/tuanvutest/Moisture	Moisture_emx	broker.emqx.io	mqtt
11	Moisture_opc	ns=2;i=4		opc.tcp://127.0.0.1:12345	opc_ua

Defining new input variables

Choose option “NEW_INPUT” to bring up new section



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.

For example, the value for Temperature_fiware variable is located in domain orion:1026 with entity id urn:ngsi-ld:test:Device:Temperature, the json file has format:

```
{
  "id": "urn:ngsi-ld:test:Device:Temperature",
  "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:busbar-789"
  },
  "https://smart-data-models.github.io/data-models/terms.jsonld#/definitions
    /category": {
    "type": "Property",
    "value": "sensor"
  },
  "value": {
    "type": "Property",
    "value": 9.06,
    "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 value “9.06”, so the path to the value is “value/value”. Thus, the inputs for the variable is

Input name: Temperature_fiware

Choose your adapter: Fiware

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

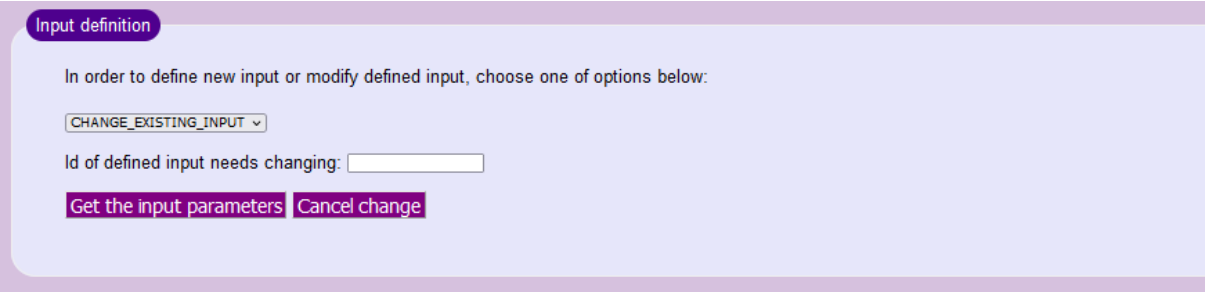
Path to data for input assignment inside data cluster: value/value

Server where input data is stored (OPC-UA, MQTT) or url (Fiware): orion:1026

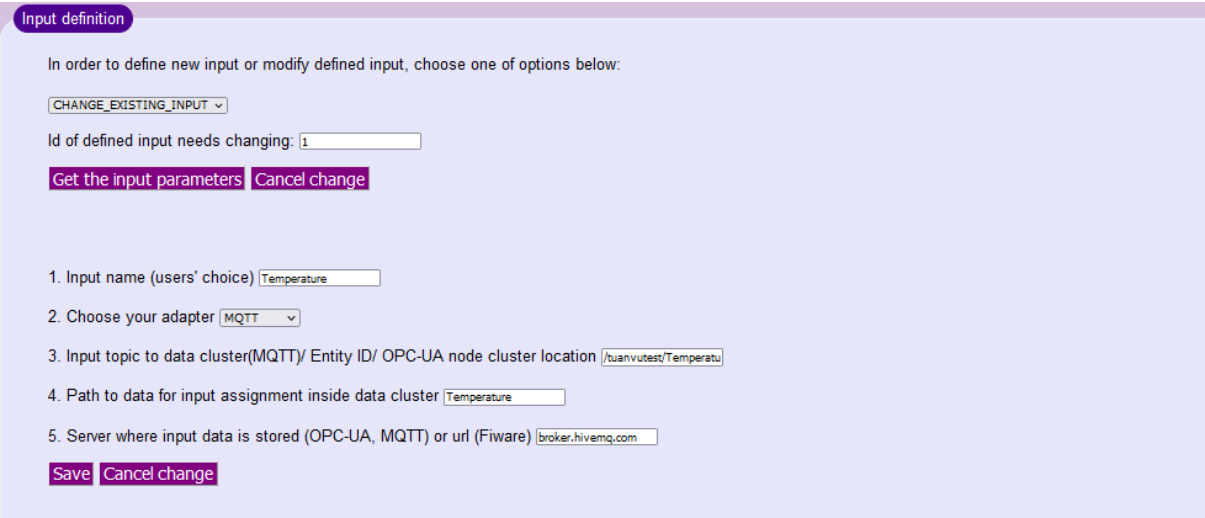
To save the variable, choose “add newly defined variable”, or else, choose “cancel change”

Modify existing input

To modify existing variable, choose “CHANGE_EXISTING_INPUT”

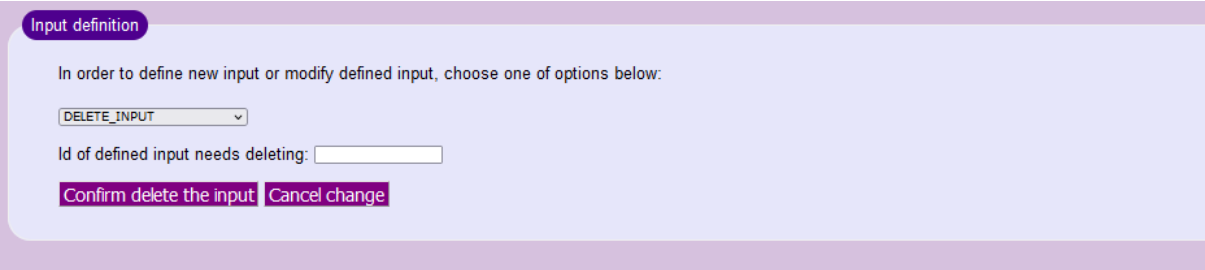


The id of the variable need to be provided, after confirming the id, the input variable detail is loaded in the windows



Delete existing input

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



Legacy data interface

The interface allows users to

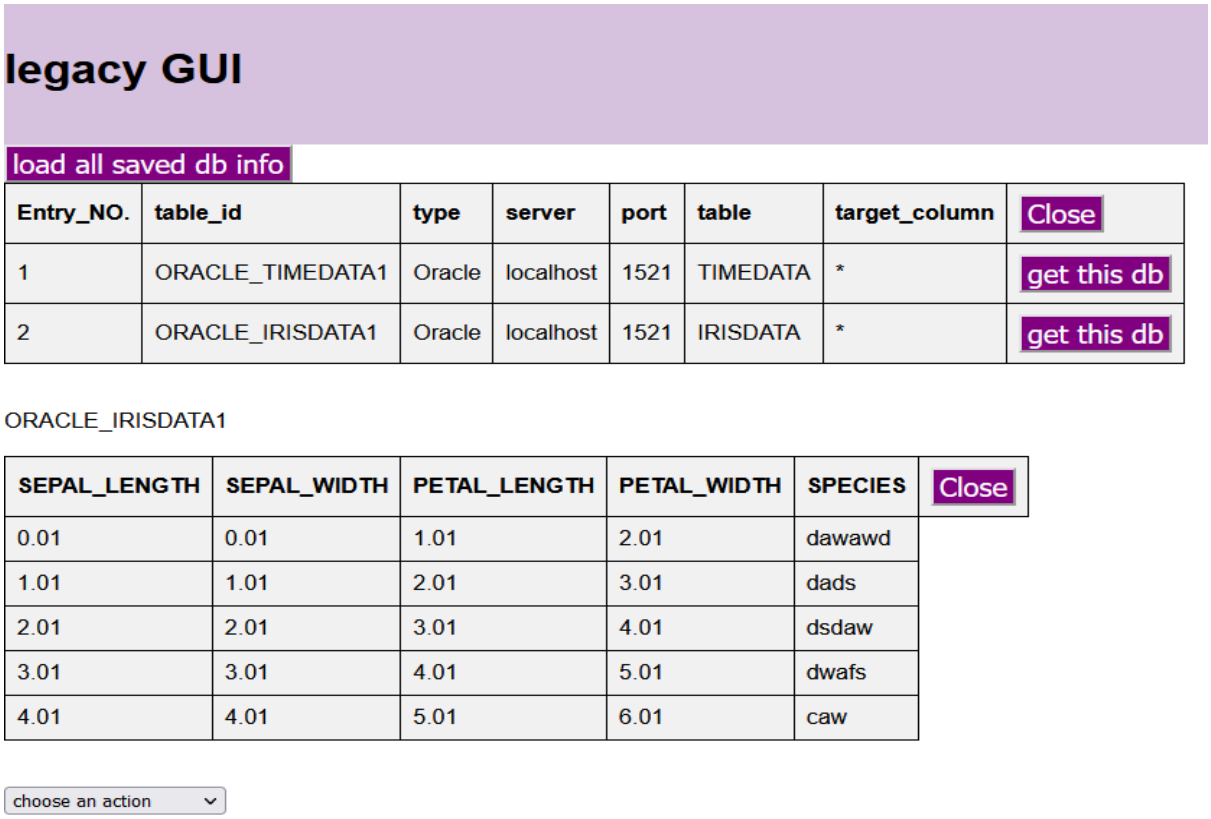
- Define database entry with parameters for saving the data within DCF (applies for oracle and sap hana)
- Define fiware entities for data extraction from saved database (the same functionality is shared with mongoDB interface)

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



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

load all saved db info

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

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).

load all saved db info

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

new database entry

Oracle

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

save database info cancel

Click “save database info” and retrieve the data

load all saved db info

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

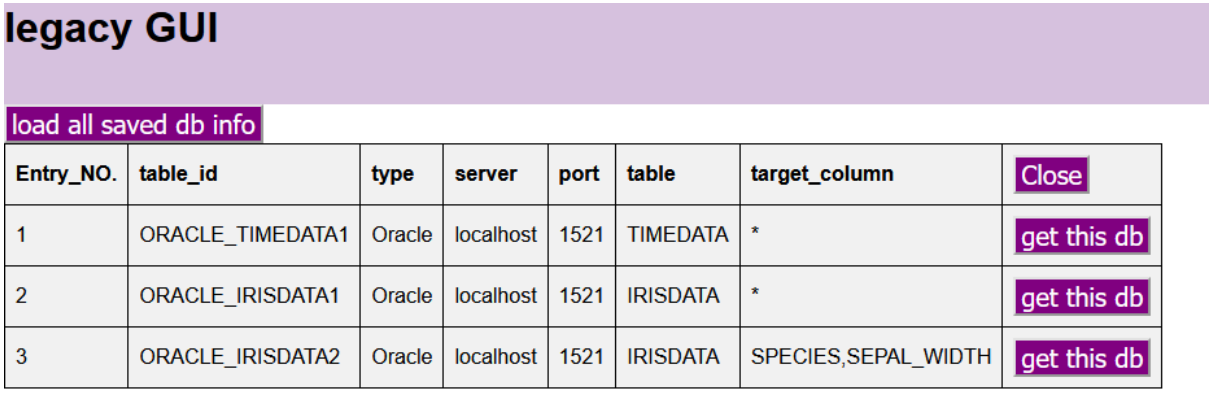
ORACLE_IRISDATA2

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

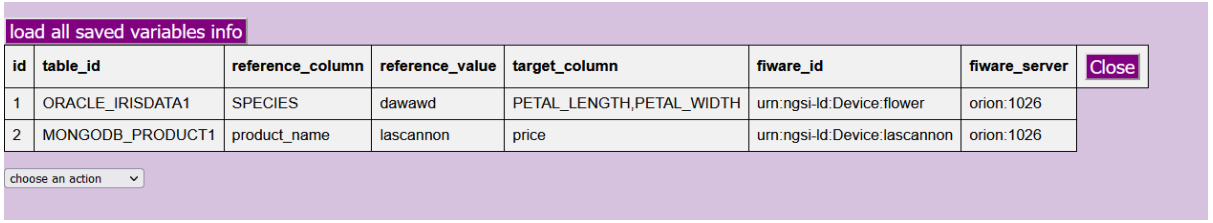
choose an action

Deleting existing entry

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

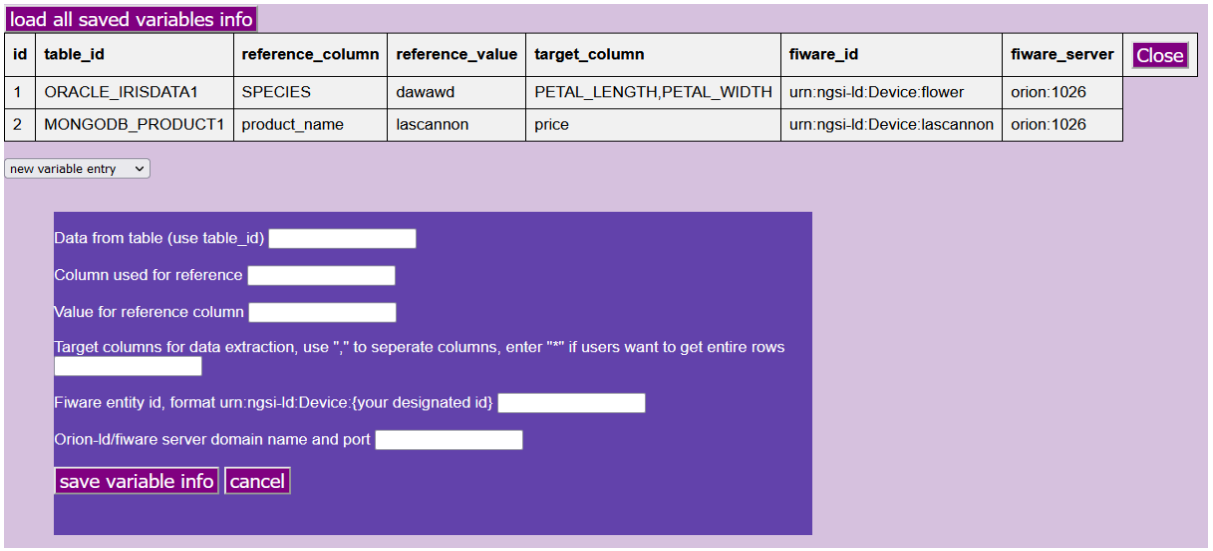


Adding new parameters for retrieving data from saved database entries or deleting existing entries



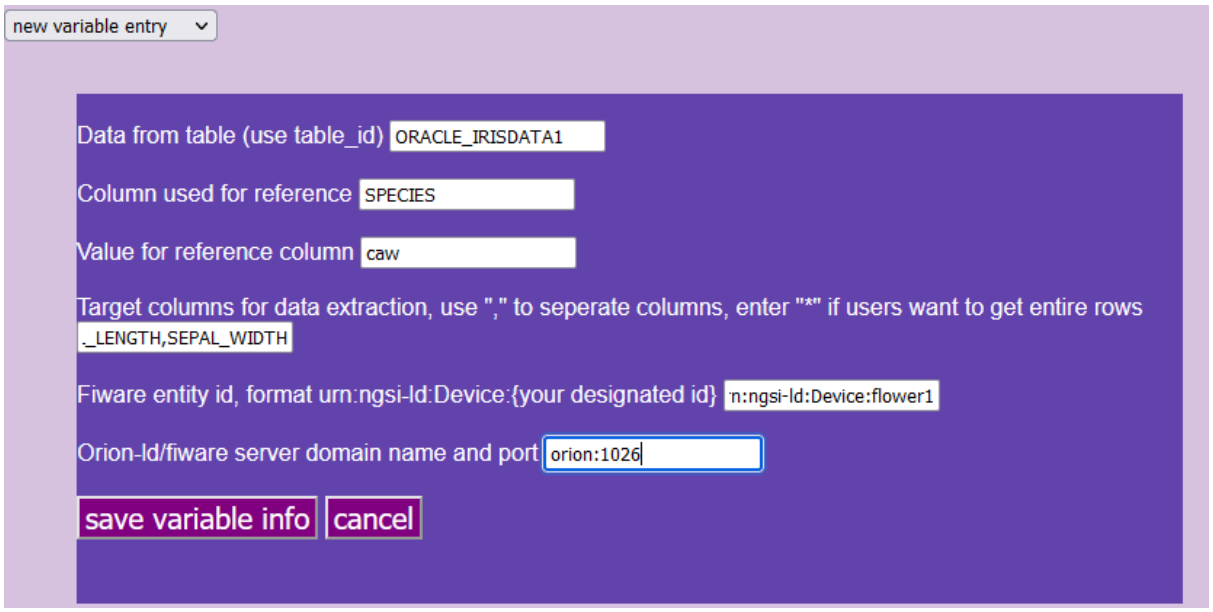
Adding new entry for retrieving data

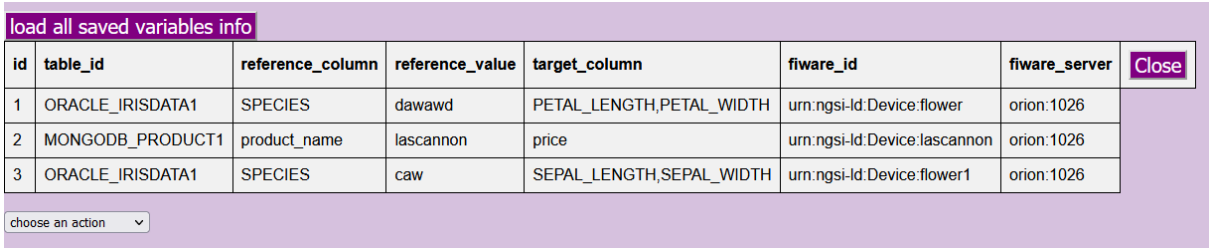
Data from saved database table can be extracted and sent through fiware as entity, this functionality is shared and can be shown in both MongoDB interface and legacy interface, to add new parameters entry, choose “new variable entry”



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

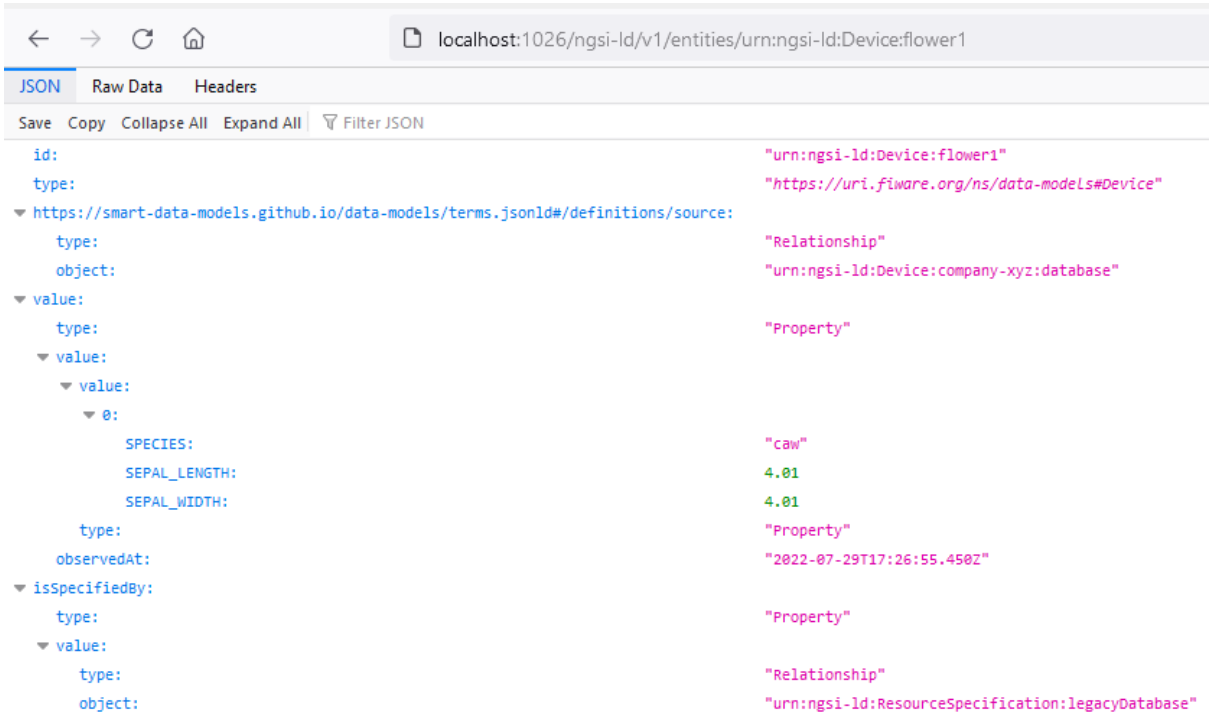
For example, defining a new entity with id “urn:ngsi-ld:Device:flower1” that retrieves the SEPAL_LENGTH, SEPAL_WIDTH of SPECIES with name “caw” from table ORACLE_IRISDATA1





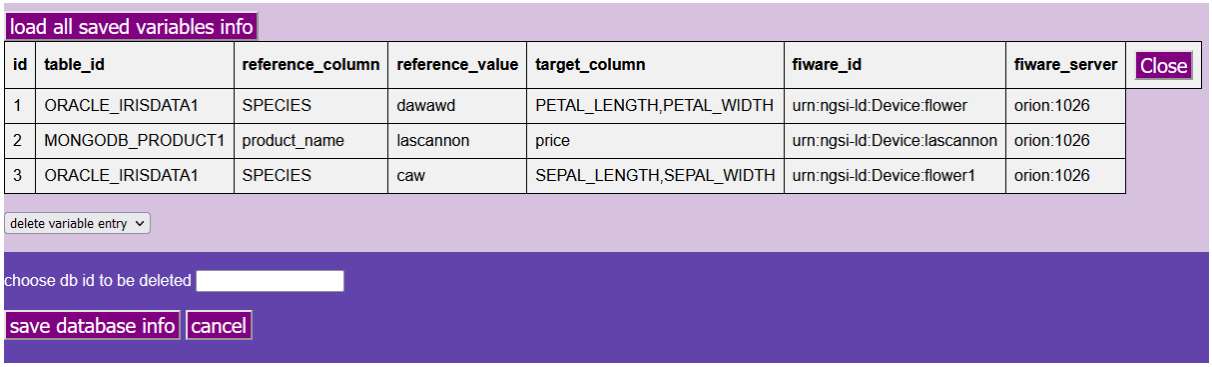
id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	orion:1026
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	orion:1026
3	ORACLE_IRISDATA1	SPECIES	caw	SEPAL_LENGTH,SEPAL_WIDTH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

Result from orion server



Delete entry

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

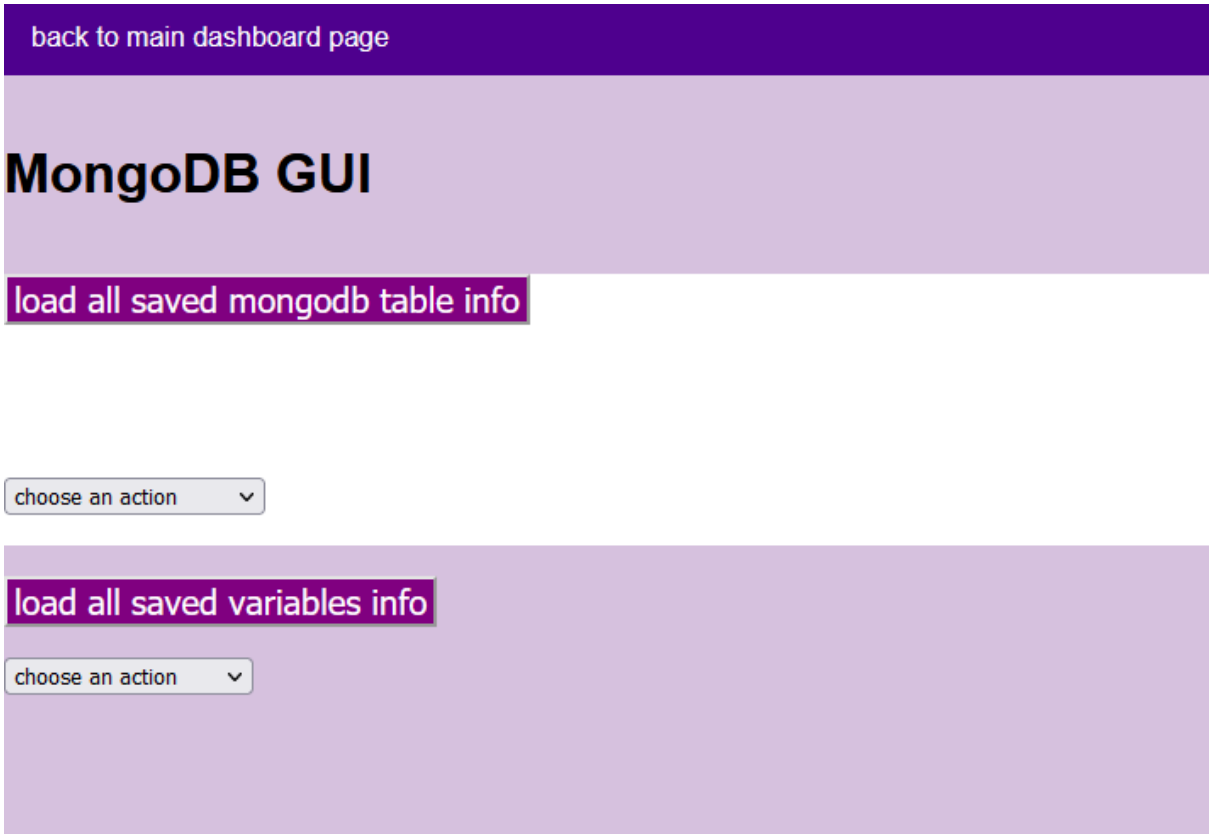


id	table_id	reference_column	reference_value	target_column	fiware_id	fiware_server
1	ORACLE_IRISDATA1	SPECIES	dawawd	PETAL_LENGTH,PETAL_WIDTH	urn:ngsi-ld:Device:flower	orion:1026
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-ld:Device:lascannon	orion:1026
3	ORACLE_IRISDATA1	SPECIES	caw	SEPAL_LENGTH,SEPAL_WIDTH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

MongoDB database interface

MongoDB database interface serves two purposes:

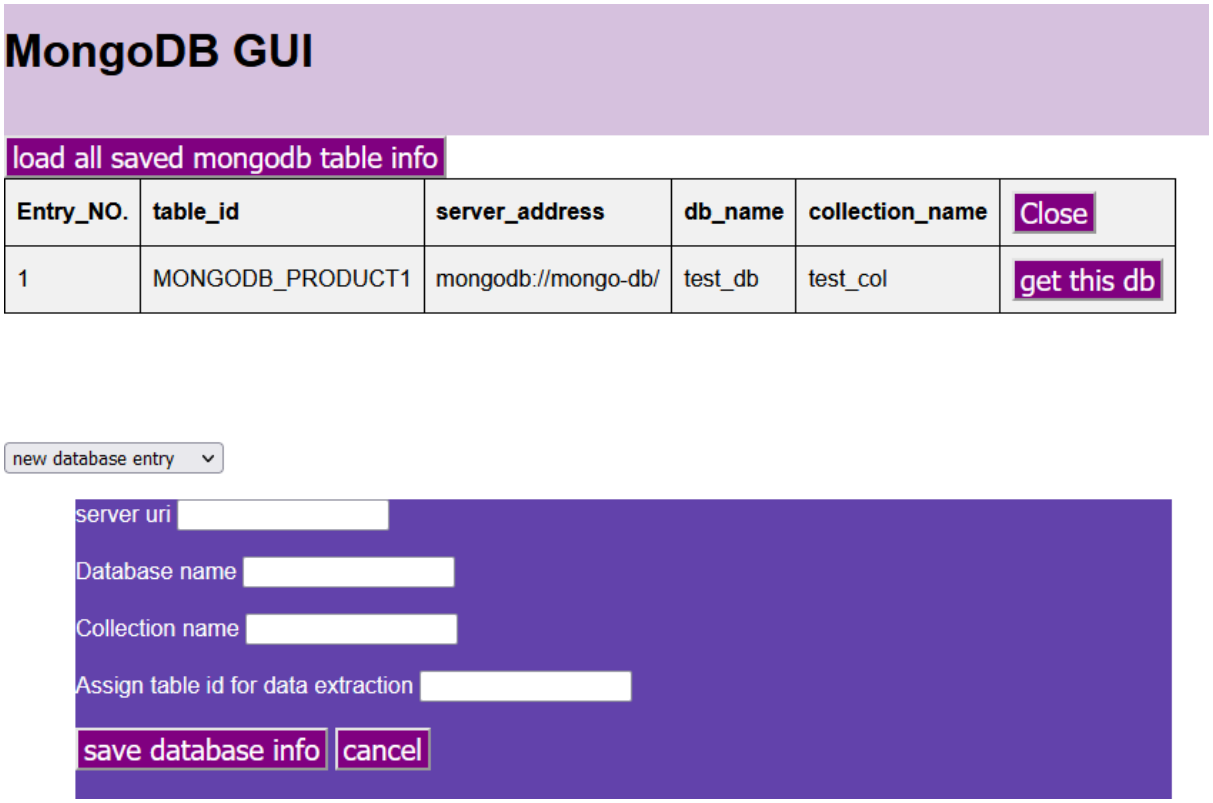
- Define database entry with parameters for saving the data within DCF (only apply for mongoDB)
- Define fiware entities for data extraction from saved database (this functionality is the same as the one in legacy interface)



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

Adding new entry

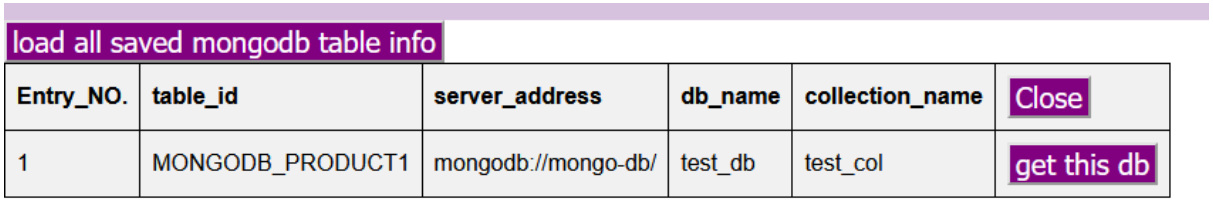
To add new entry, select “new database entry”



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

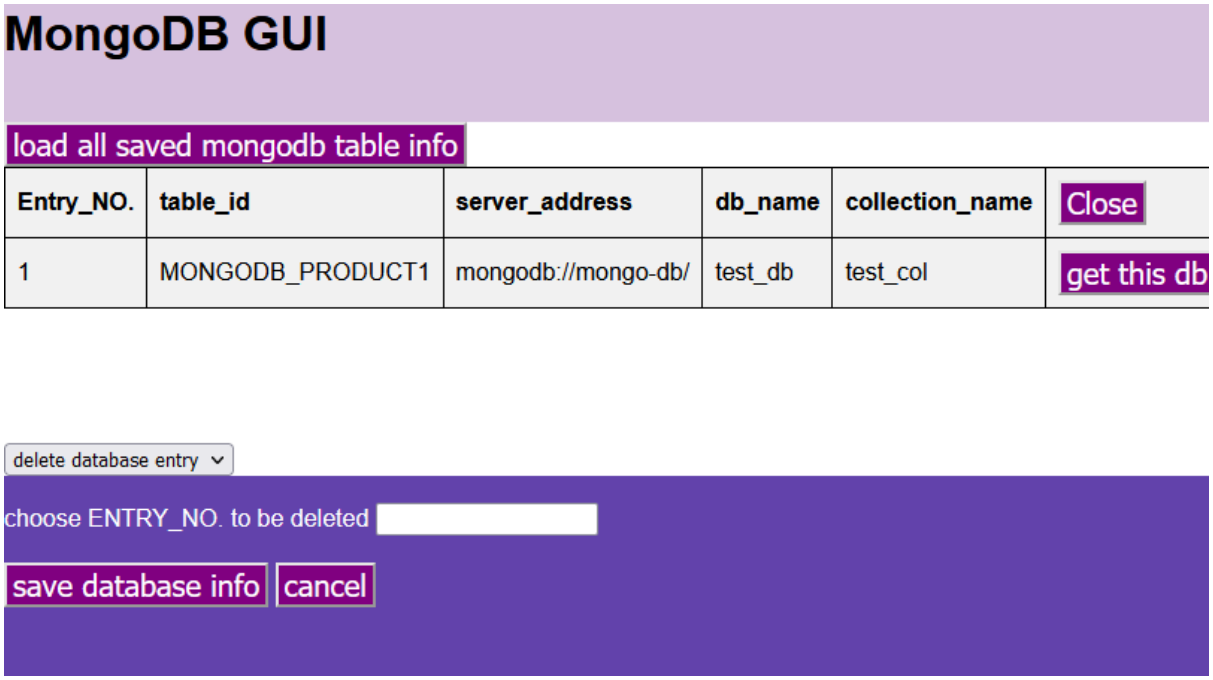


MONGODB_PRODUCT1

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



Adding new parameters entry for data retrieval or deleting existing entry

The procedures are the same as in legacy interface