

# DCF

## Data Collection Framework

### User Manual



## Contents

INTRODUCTION .....	3
DCF ARCHITECTURE .....	4
DOWNLOADS ADDITIONAL FILES AND INSTALLATION .....	5
USER INTERFACE .....	7
Information/historical data dashboard .....	7
Event processing interface .....	8
Creating new operation .....	9
Modifying existing operation .....	12
Deleting existing operation .....	14

## 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.

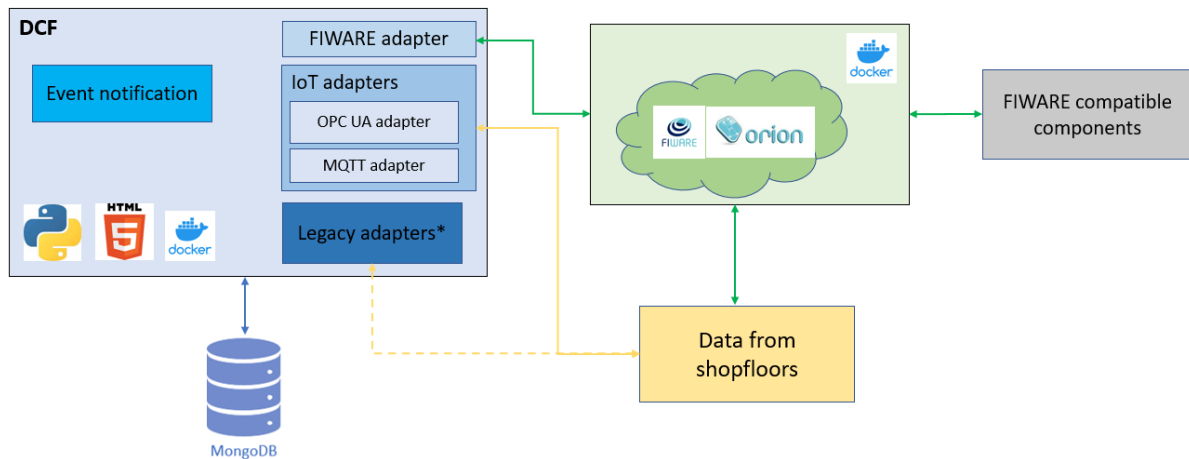


Figure 1: DCF architecture

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

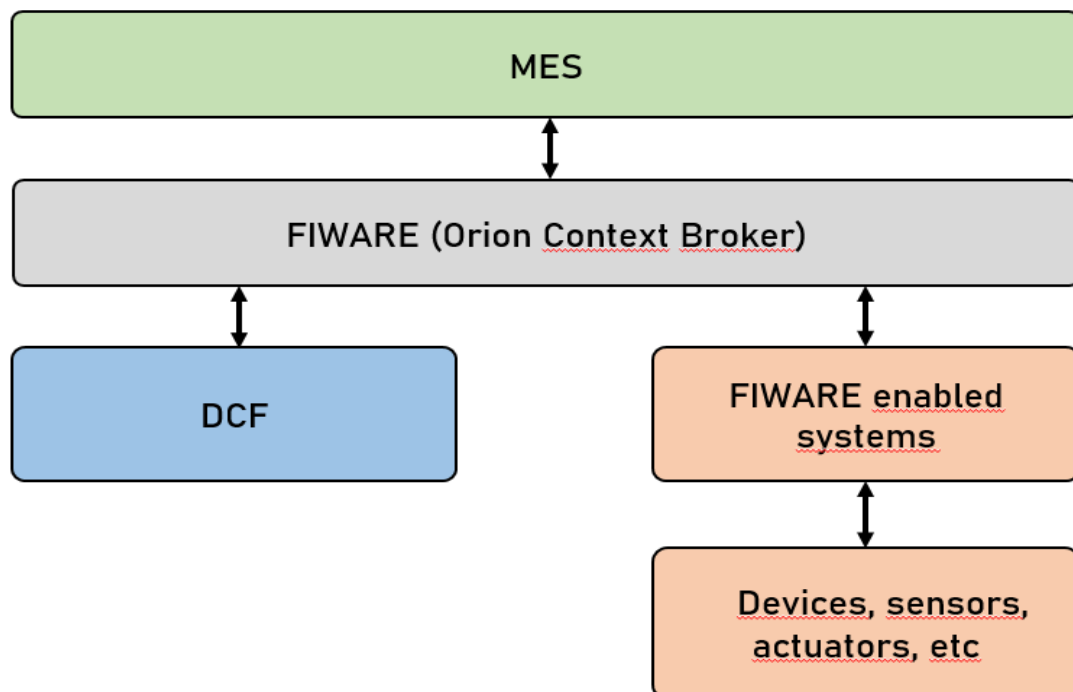


Figure 2: DCF communication with other components

## DOWNLOADS ADDITIONAL FILES AND INSTALLATION

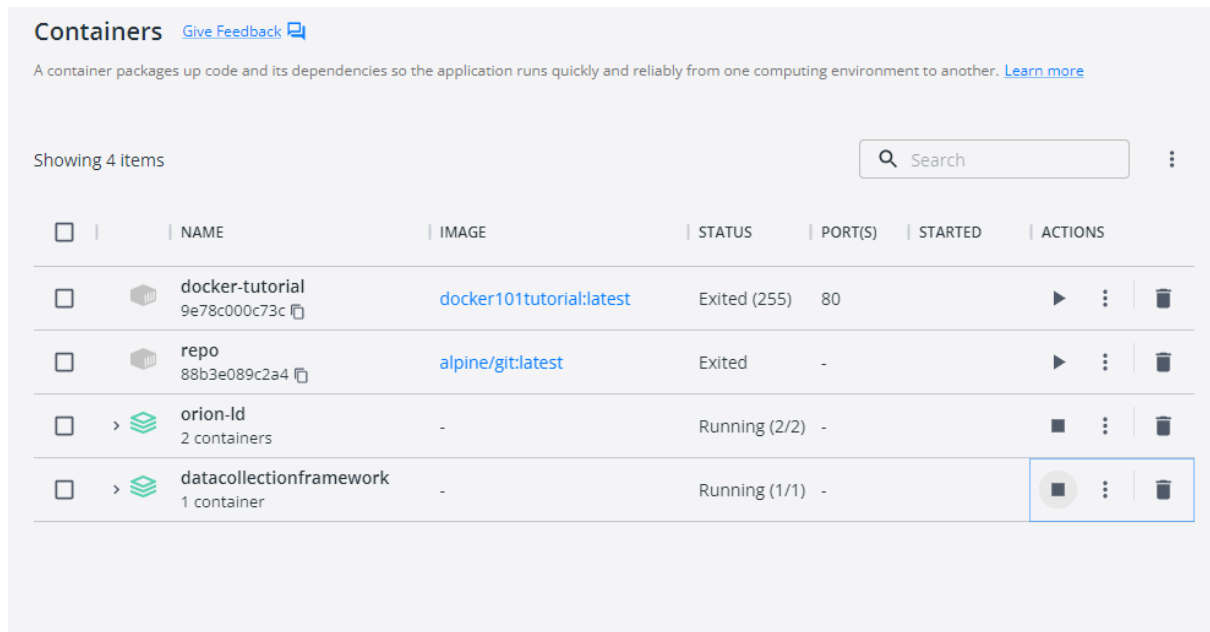
Beside DCF image, only docker-compose file is needed to compose the image into container. To get the docker-compose.yml file, download the file from <https://github.com/TAU-FASTLab/DCF>. Version of DCF can be changed to compose the suitable version.



```

docker-compose.yml - Notepad
File Edit Format View Help
version: "3.9"
services:
  dcf:
    image: dcf:1.6
    container_name: dcf
    hostname: dcf
    ports:
      - "1028:1028"
    expose:
      - "1028"
networks:
  default:
    name: orion-ld_default
    external: true
  
```

Before composing the image, the networks parameter in docker-compose.yml file needs to be configured.



**Containers** [Give Feedback](#)

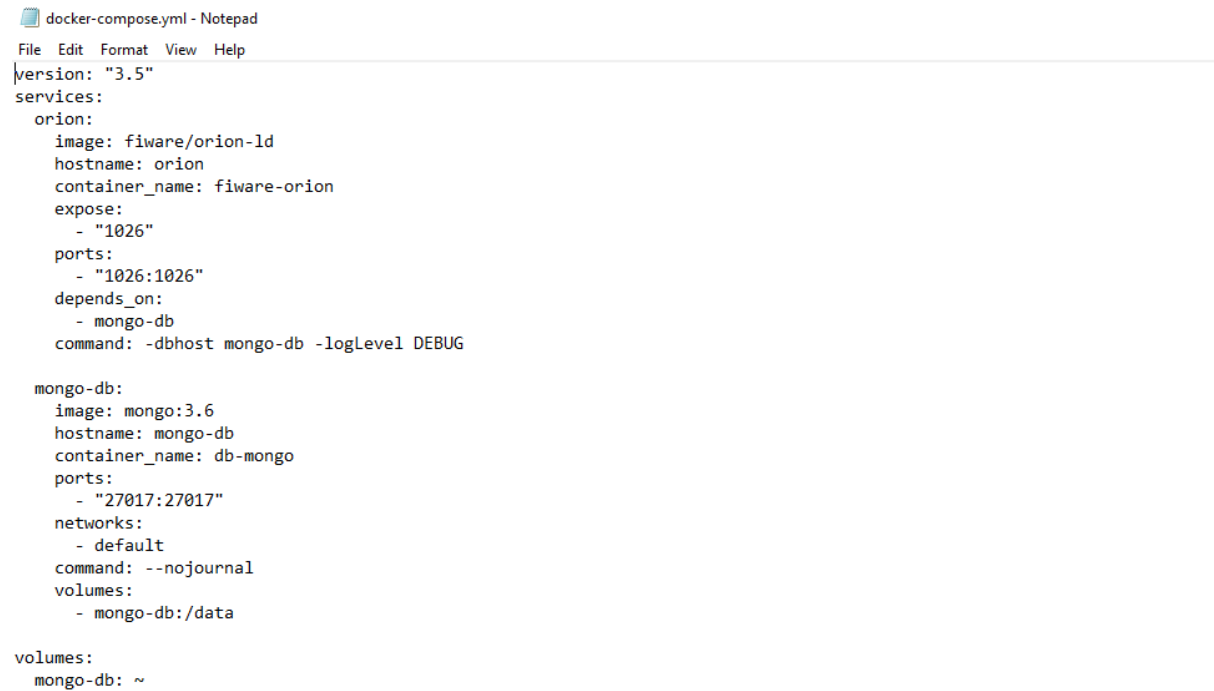
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Showing 4 items

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	docker-tutorial 9e78c000c73c	docker101tutorial:latest	Exited (255)	80		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	repo 88b3e089c2a4	alpine/git:latest	Exited	-		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	orion-ld 2 containers	-	Running (2/2)	-		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	datacollectionframework 1 container	-	Running (1/1)	-		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

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. To find the name of the network, users need to find the docker-compose.txt file of the fiware container.

## DCF manual



```
docker-compose.yml - Notepad
File Edit Format View Help
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

  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: ~
```

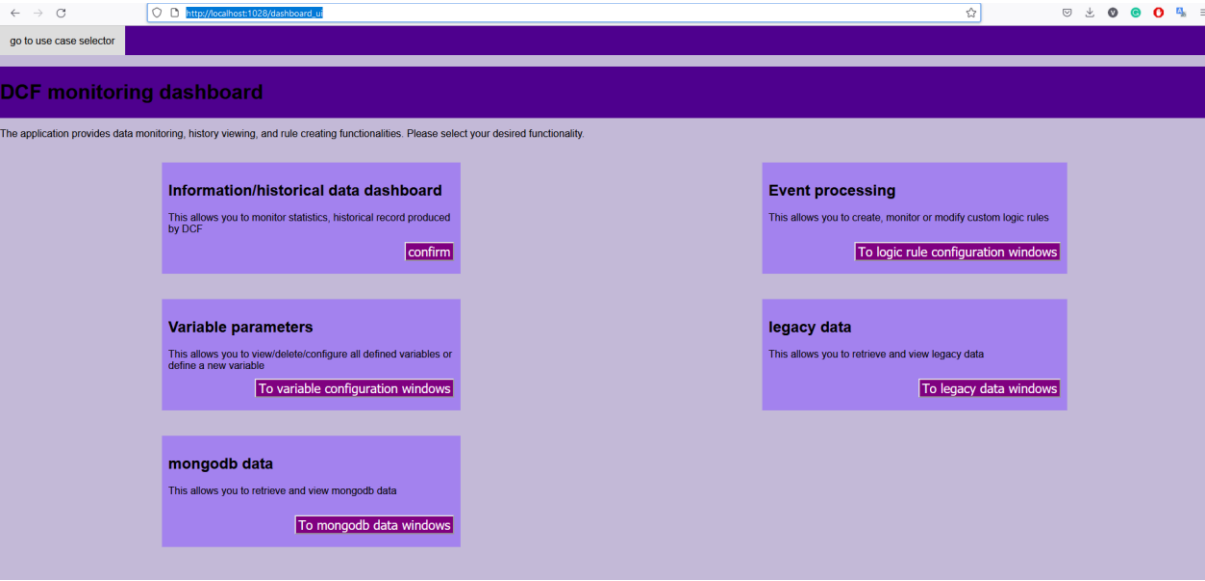
In this docker-compose.txt file, the “network” part is “default”, name of container is “orion-ld”, so the name of the network that needs to be defined in the DCF docker-compose.txt file is “orion-ld\_default”

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) or simply want to use fiware IP instead of using domain name of containerized Fiware server, 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.

back to main dashboard page					
DCF history monitor					
load all recorded event data					
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	<a href="#">Close</a>
1	ALARM	Pressure > 100 and Temperature > 30	1	fiware,orion:1026,urn:ngsi-Id:test:Alert:alert1	<a href="#">Close</a>
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-Id: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-Id:test:Alert:count20	
9	COUNT	Pressure_fiware >= 20		fiware,orion:1026,urn:ngsi-Id:test:Alert:count100	
10	COUNT	Pressure > 100		mqtt,broker.hivemq.com,/tuanvutest/counter10000	

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	<a href="#">Close</a>
1	Temperature	/tuanvutest/Temperature	Temperature	broker.hivemq.com	<a href="#">Close</a>
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-Id:test:Device:Temperature	value/value	orion:1026	
9	Pressure_fiware	urn:ngsi-Id: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	



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

5	ALARM	Temperature < 1000	mqtt.broker.hivemq.com/tuanvutest/alarmlu
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

**Available inputs**

In order to view the list of available inputs click the load button. [Load](#)

**Operation definition**

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

CHOOSE AN OPTION

CHOOSE AN OPTION

NEW\_FUNCTION

CHANGE\_EXISTING\_FUNCTION

DELETE\_FUNCTION

## Creating new operation

**Operation definition**

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

NEW\_FUNCTION

All operation parameters 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 operation 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		

3. Choose a desired variable from the list of defined variables **Temperature** [Insert variable](#)
4. Users can input numeric data, use dot to separate decimals [Insert number](#)
5. Select the desired logic operators from the list below.

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

True

[Save](#) [Cancel all changes](#)

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-Id, 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-Id:test:Alert:count5”, on my containerized fiware/orion-Id server so the server should be orion:1026

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

NEW\_FUNCTION

All operation parameters 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 operation 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-Id	orion:1026	ngsi-Id:test:Alert:count5

3. Choose a desired variable from the list of defined variables **Temperature**
4. Users can input numeric data, use dot to separate decimals
5. Select the desired logic operators from the list below.

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

Temperature\_fiware > 5 and Pressure\_fiware <= 30

**Save** **Cancel all changes**

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

click load all defined operations button to show updated operations list and their parameters **load all defined operations**

id	Function_type	Logic_condition	Alarm_limit_count	Result_published_to	Close
1	ALARM	Pressure > 100 and Temperature > 30	1	mqtt,broker.hivemq.com,/tuanvutest/alarm1,fiware,orion,urn:ngsi-Id: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-Id: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-Id:test:Alert:count20	
9	COUNT	Pressure_fiware >= 20		fiware,localhost,urn:ngsi-Id:test:Alert:count100	
10	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30		fiware,orion,urn:ngsi-Id: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

← → ↺ http://localhost:1028/DCF\_monitor\_event

back to main dashboard page

DCF history monitor

load all recorded event data

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

← → ↺ http://localhost:1026/ngsi-ld/v1/entities/urn:ngsi-ld:test.Alert:count5

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

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:	"Property"
type:	"alert"
https://smart-data-models.github.io/data-models/terms.jsonld#definitions/validTo:	"Property"
type:	"DateTime"
value:	"2022-07-29T11:14:33.682"
value:	"Property"
type:	"1"
observedAt:	"2022-07-29T11:14:33.680Z"
description:	"Property"
type:	"COUNT;Temperature_fiware > 5 and Pressure_fiware <= 30;;fiware,orion,urn:ngsi-ld:test:Alert:count5"
https://smart-data-models.github.io/data-models/terms.jsonld#definitions/dateIssued:	"Property"
type:	"Property"
value:	"DateTime"
@type:	"2022-07-29T11:14:33.682"
@value:	"2022-07-29T11:14:33.682"
https://smart-data-models.github.io/data-models/terms.jsonld#definitions/alertSource:	"Relationship"
type:	"urn:ngsi-ld:dcf-logic-engine"
object:	"Property"
https://smart-data-models.github.io/data-models/terms.jsonld#definitions/validFrom:	"Property"
type:	"DateTime"
value:	"2022-07-29T11:14:33.682"
@type:	"Property"
@value:	"high"
https://smart-data-models.github.io/data-models/terms.jsonld#definitions/severity:	"Property"
type:	"high"
value:	"false"
humanVerified:	"Property"
type:	"false"
value:	

The operation is registered and output is published.

## Modifying existing operation

To modifying existing operation, select “CHANGE\_EXISTING\_FUNCTION” option

**Operation definition**

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:

**Fetch the operation parameters**

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

**Operation definition**

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:

**Fetch the operation parameters**

All operation parameters 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 operation 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	<b>MQTT</b> ▾	broker.hivemq.com	log-id-test:Alert:count5 <input type="button" value="X"/>

3. Choose a desired variable from the list of defined variables **Temperature** ▾
4. Users can input numeric data, use dot to separate decimals
5. Select the desired logic operators from the list below.

**( ) < > == >= <= != and or \*\* \* = + / not**

New condition is added

Operation definition

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: 10

Fetch the operation parameters

All operation parameters 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 operation 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	MQTT	broker.hivemq.com	/tuanvutest/counter100

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

Temperature

Insert variable

4. Users can input numeric data, use dot to separate decimals

Insert number

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

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

Temperature\_fiware > 5 and Pressure\_fiware <= 30 and Pressure\_fiware > 0

Save

Cancel all changes

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.

back to main dashboard page

DCF history monitor

load all recorded event data

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”

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: 9

Confirm delete the functionCancel

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,um.ngsi-Id.test.Alert.alert1	
2	COUNT	Pressure < 80 and Pressure > 50		mqtt.broker.hivemq.com,/tuavutest/counter	
3	COUNT	Temperature > 20		mqtt.broker.hivemq.com,/tuavutest/counter500,fiware.orion:1026,um.ngsi-Id.test.Alert.cnt1	
4	COUNT	Temperature > 4		mqtt.broker.hivemq.com,/tuavutest/counter3	
5	ALARM	Temperature < 1000	3	mqtt.broker.hivemq.com,/tuavutest/alarm10	
6	COUNT	Pressure_emx > 10		mqtt.broker.hivemq.com,/tuavutest/count12	
7	COUNT	Moisture_emx > 10 and Moisture > 10		mqtt.broker.hivemq.com,/tuavutest/count372	
8	COUNT	Temperature_fiware <= 20		fiware.orion:1026,um.ngsi-Id.test.Alert.count20	
9	COUNT	Temperature_fiware > 5 and Pressure_fiware <= 30 and Pressure_fiware > 0		mqtt.broker.hivemq.com,um.ngsi-Id.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

Back to DCF monitoring dashboard

### Input parameters

This interface is used to define new inputs or monitor/ modify existing inputs. Defined inputs are used to create/modify logic operation using [Event definition and processing](#) interface

**Input overview**

In order to view all defined inputs, select load button **Load**

Id	Name	Topic to data cluster/ Fiware id/ Opc-ua data node	Path to variable data	Server	Adapter_type	Close
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-Id:test:Device:Temperature	value/value	orion:1026	fiware	
9	Pressure_fiware	urn:ngsi-Id: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	

**Input definition**

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

NEW\_INPUT  
 CHOOSE AN OPTION  
 NEW\_INPUT  
 CHANGE\_EXISTING\_INPUT  
 DELETE\_INPUT

## Defining new input variables

Choose option “NEW\_INPUT” to bring up new section

**Input definition**

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

NEW\_INPUT

1. Input name (users' choice)
2. Choose your adapter **No adapter**
3. Input topic to data cluster(MQTT)/ Entity ID/ OPC-UA node cluster location
4. Path to data for input assignment inside data cluster
5. Server where input data is stored (OPC-UA, MQTT) or url (Fiware)

**Save** **Cancel change**

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:

```

1 {
2   "id": "urn:ngsi-ld:test:Device:Temperature",
3   "type": "https://uri.fiware.org/ns/data-models#Device",
4   "https://smart-data-models.github.io/data-models/terms.jsonld#/definitions
   /source": {
5     "type": "Relationship",
6     "object": "urn:ngsi-ld:Device:company-xyz:busbar-789"
7   },
8   "https://smart-data-models.github.io/data-models/terms.jsonld#/definitions
   /category": {
9     "type": "Property",
10    "value": "sensor"
11  },
12  "value": {
13    "type": "Property",
14    "value": 9.06,
15    "observedAt": "2020-12-01T11:23:19.000Z"
16  },
17  "https://smart-data-models.github.io/data-models/terms.jsonld#/definitions
   /deviceState": {
18    "type": "Property",
19    "value": "ok"
20  },
21  "isSpecifiedBy": {
22    "type": "Property",
23    "value": {
24      "type": "Relationship",
25      "object": "urn:ngsi-ld:ResourceSpecification:company-xyz:sensor"
26    }
27  }
28 }

```

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 screenshot shows a window titled 'Input definition' with a light purple background. At the top, a dark purple pill-shaped button contains the text 'Input definition'. Below this, a text label reads: 'In order to define new input or modify defined input, choose one of options below:'. A dropdown menu is set to 'CHANGE\_EXISTING\_INPUT'. Below the dropdown, a text label reads: 'Id of defined input needs changing:'. To the right of this label is an empty text input field. At the bottom, there are two purple buttons: 'Get the input parameters' and 'Cancel change'.

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

The screenshot shows the same 'Input definition' window. The dropdown menu is still 'CHANGE\_EXISTING\_INPUT'. The text input field for 'Id of defined input needs changing:' now contains the number '1'. Below this, there are five numbered steps, each with a text input field: 1. 'Input name (users' choice)' contains 'Temperature'; 2. 'Choose your adapter' has a dropdown menu set to 'MQTT'; 3. 'Input topic to data cluster(MQTT)/ Entity ID/ OPC-UA node cluster location' contains '/tuanvutest/Temperatu'; 4. 'Path to data for input assignment inside data cluster' contains 'Temperature'; 5. 'Server where input data is stored (OPC-UA, MQTT) or url (Fiware)' contains 'broker.hivemq.com'. At the bottom, there are two purple buttons: 'Save' and 'Cancel change'.

### Delete existing input

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

The screenshot shows the 'Input definition' window. The dropdown menu is now set to 'DELETE\_INPUT'. The text label below it reads: 'Id of defined input needs deleting:'. To the right of this label is an empty text input field. At the bottom, there are two purple buttons: 'Confirm delete the input' and 'Cancel change'.

## 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

## legacy GUI

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

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	

choose an action ▼

*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

new database entry

Oracle

server address

server port

Username

Password

Table name

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

Assign table id for data extraction

save database info

cancel

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

## legacy GUI

## 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

delete database entry ▾

choose ENTRY\_NO. to be deleted 

save database info

cancel

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

## load all saved variables info

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-Id:Device:flower	orion:1026	
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-Id:Device:lascannon	orion:1026	

choose an action ▾

## 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”

## load all saved variables info

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-Id:Device:flower	orion:1026	
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-Id:Device:lascannon	orion:1026	

new variable entry ▾

Data from table (use table\_id) Column used for reference Value for reference column 

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

Fiware entity id, format urn:ngsi-Id:Device:{your designated id} Orion-Id/fiware server domain name and port 

save variable info

cancel

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-lid:Device:flower1” that retrieves the SEPAL\_LENGTH, SEPAL\_WIDTH of SPECIES with name “caw” from table ORACLE\_IRISDATA1

new variable entry

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 separate columns, enter "\*" if users want to get entire rows

\_LENGTH,SEPAL\_WIDTH

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

n:ngsi-lid:Device:flower1

Orion-lid/fiware server domain name and port

orion:1026

save variable info

cancel

load all saved variables info

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-lid:Device:flower	orion:1026	
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-lid:Device:lascannon	orion:1026	
3	ORACLE_IRISDATA1	SPECIES	caw	SEPAL_LENGTH,SEPAL_WIDTH	urn:ngsi-lid:Device:flower1	orion:1026	

choose an action

Result from orion server

←

→

↺

🏠

localhost:1026/ngsi-lid/v1/entities/urn:ngsi-lid:Device:flower1

JSONRaw DataHeaders

SaveCopyCollapse AllExpand AllFilter JSON

id:

type:

▼ https://smart-data-models.github.io/data-models/terms.jsonld#/definitions/source:

type:

object:

▼ value:

type:

▼ value:

▼ value:

▼ 0:

SPECIES:

SEPAL\_LENGTH:

SEPAL\_WIDTH:

type:

observedAt:

▼ isSpecifiedBy:

type:

▼ value:

type:

object:

"urn:ngsi-lid:Device:flower1"

"https://uri.fiware.org/ns/data-models#Device"

"Relationship"

"urn:ngsi-lid:Device:company-xyz:database"

"Property"

"caw"

4.01

4.01

"Property"

"2022-07-29T17:26:55.450Z"

"Property"

"Relationship"

"urn:ngsi-lid:ResourceSpecification:legacyDatabase"

Delete entry

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

load all saved variables info

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-lid:Device:flower	orion:1026	
2	MONGODB_PRODUCT1	product_name	lascannon	price	urn:ngsi-lid:Device:lascannon	orion:1026	
3	ORACLE_IRISDATA1	SPECIES	caw	SEPAL_LENGTH,SEPAL_WIDTH	urn:ngsi-lid:Device:flower1	orion:1026	

delete variable entry ▼

choose db id to be deleted

save database infocancel



## 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

load all saved mongodb table info

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

new database entry

server uri

Database name

Collection name

Assign table id for data extraction

save database info

cancel

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

load all saved mongodb table info

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

load all saved mongodb table info

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

save database info

cancel

[Adding new parameters entry for data retrieval or deleting existing entry](#)

The procedures are the same as in legacy interface