Connecting Cloudino Connector to FIWARE IoT

1. What is FIWARE IoT

FIWARE is an open software ecosystem provided by the FIWARE Community (htttp://www.fiware.org).

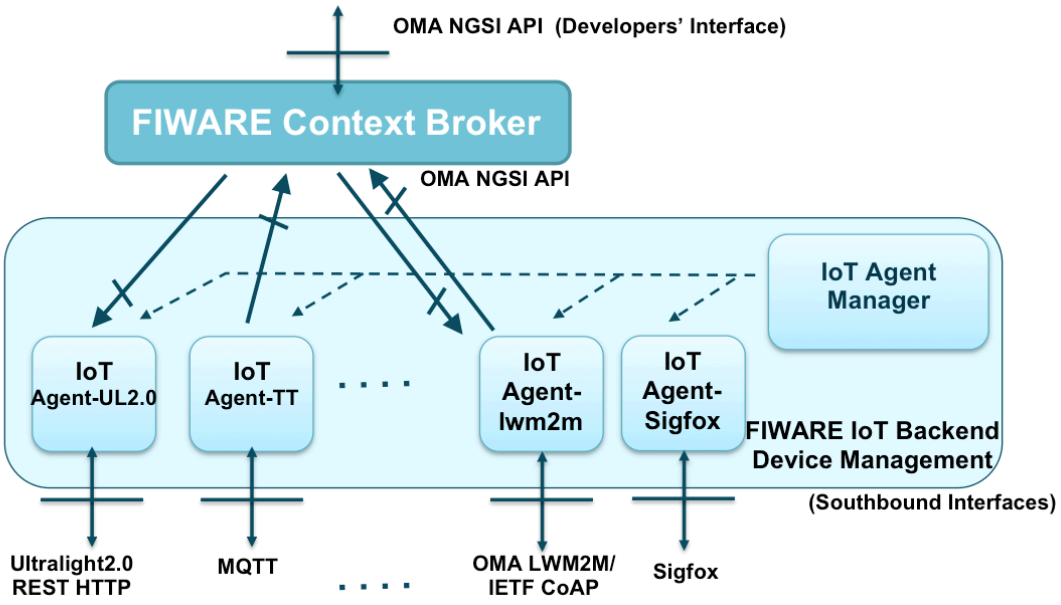
FIWARE exposes to developers Data Context elements or entities (JSON objects) with attributes and metadata with a uniform REST API (NGSI9/10). NGSI is now being adopted by more than 60 smartcities worldwide:

- http://connectedsmartcities.eu/open-agile-smart-cities

Orion Context Broker is available at: http://bit.ly/github-ContextBroker

FIWARE IoT is an opensource software stack aiming to bring Data-level interoperability to the complex salad of standards and protocols in the world of IoT today.

FIWARE IoT is able to expose -by means of the Orion Context Broker component- all IoT devices information and commands using the Data Context API (NGSI).

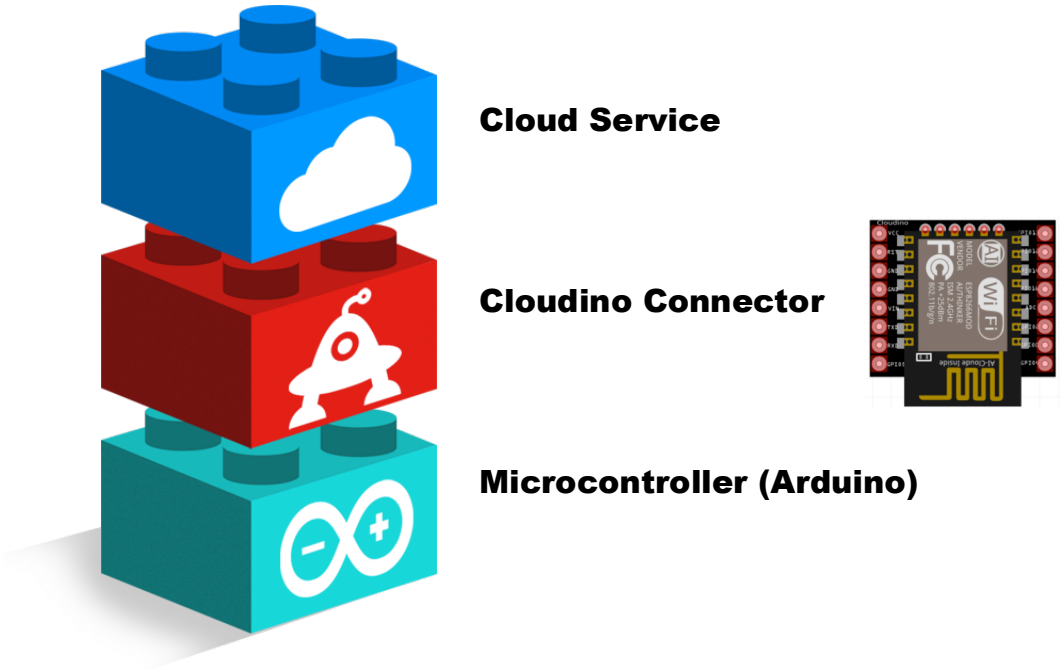


Any IoT standard or proprietary protocol can be connected to FIWARE via the IoT-Agent components. Currently FIWARE IoT is providing IoT-Agents for:

- Ultralight2.0/HTTP: http://bit.ly/fiware_iot-ul20
- MQTT/TCP: http://bit.ly/fiware_iot-ul20
- LWM2M/CoAP: http://bit.ly/fiware_iot-lwm2m-coap
- SIGFox Cloud: https://github.com/telefonicaid/sigfox-iotagent

2. What is Cloudino Connector

Cloudino Connector is a WiFi device that lets you connect to the Internet different microcontrollers (Atmel AVR, PIC, Intel Edison, etc.) in a simple and transparent way.



The **Cloudino Connector** is not like an Arduino shield, is other independent processor working in parallel dedicated only to the network layer including the IoT protocols, leaving the Arduino or microcontroller dedicated only to process data from sensors and control actuators, while allows reprogramming the Arduino or microcontroller via WiFi or Cloud. Leaving the configuration of protocols to the Cloudino Web Interface and out off your Ardurino code.

3. How to use Cloudino Connector with FIWARE

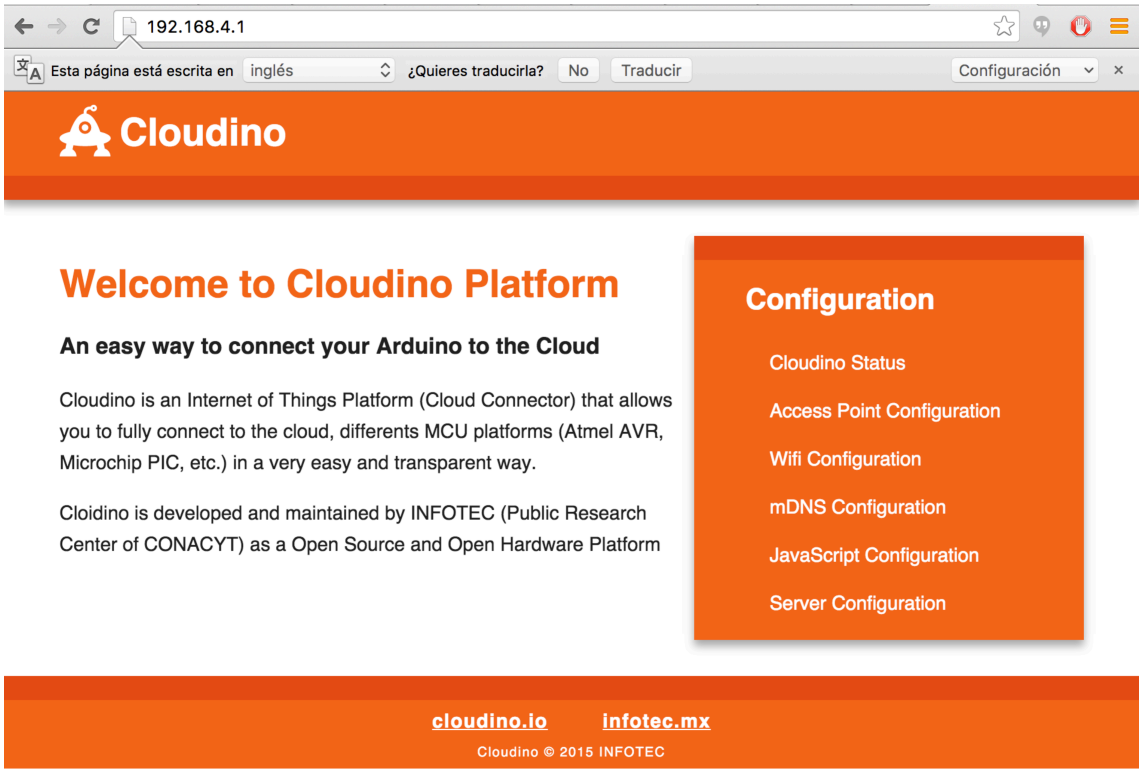
Cloudino Connector can be integrated with **FIWARE** above-described **FIWARE IoT** ecosystems using different mechanisms:

- Direct Connection
- Connection via MQTT IoT-Agent
- Connection via Cloudino.io cloud service (work in process)

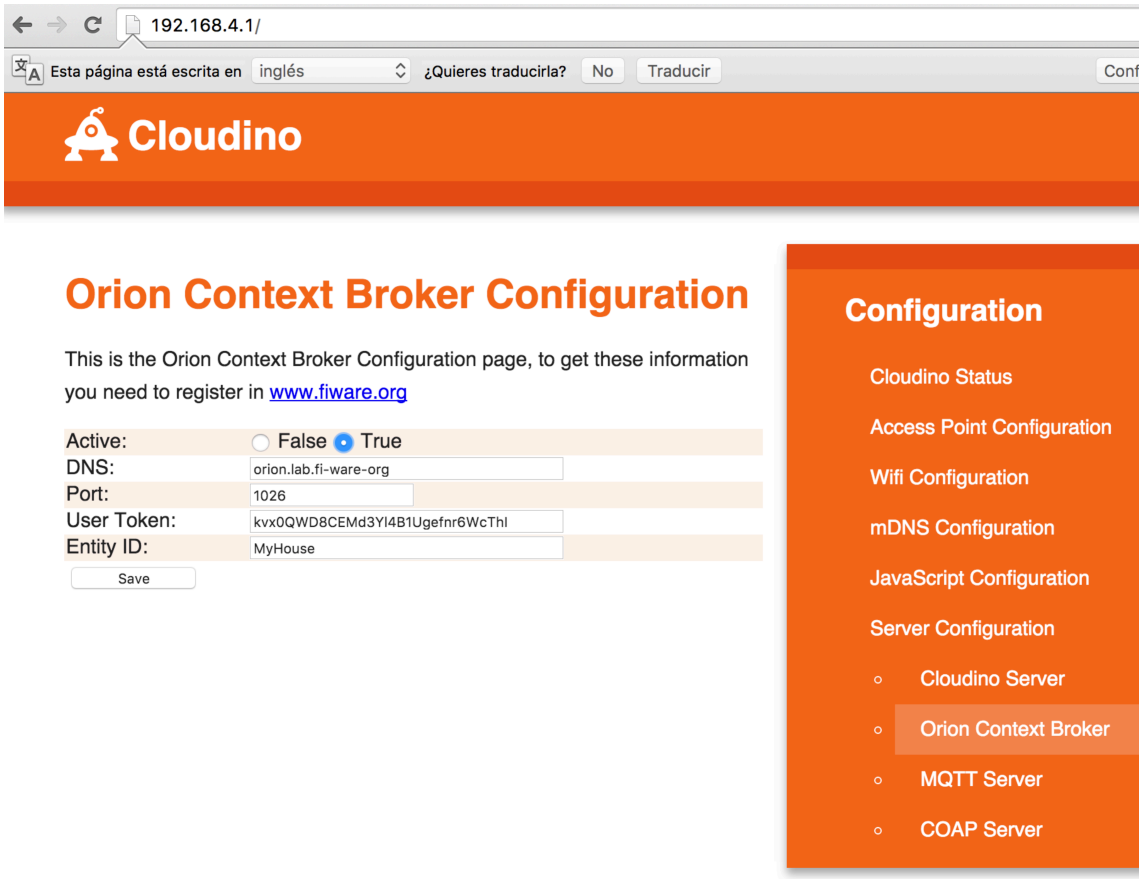
3.1. Direct Connection to FIWARE Context Broker

Cloudino Connector can connect to the FIWARE Context Broker without an IoT-Agent, using the simple Cloudino **Configuration Web Interface**.

The **Cloudino Connector** starts an access point that lets you connect to the configuration web interface at: http://192.168.4.1



To use a direct connection to FIWARE Context Broker you have to select the FIWARE Orion Context Broker in Server Configuration and setting the next fields:



Active: *True*

DNS: *Orion.lab.fi-ware.org*

Port: *1026*

User Token: *(generate token from FIWARE Server, for example kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI. For more information go to* https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Publish/Subscribe_Broker_-_Orion_Context_Broker_-_Quick_Start_for_Programmers).

Entity ID: *(Instance name, for example "MyHouse")*

Example of Arduino Code to Post Temperature and Humidity

```
#include <Cloudino.h>
#include <dht11.h>

#define DHT11PIN 8

Cloudino cdino;          //Cloudino Library
dht11 DHT11;             //DHT11 Library

void getSensor()
{
   int chk = DHT11.read(DHT11PIN);
   cdino.post("temperature",String((float)DHT11.temperature,2));
   cdino.post("humidity",String((float)DHT11.humidity,2));
   cdino.print("Timer done!");    //Send to console
}

void setup()
{
   cdino.setInterval(10000,getSensor);  //Timer every 10 seconds
   cdino.begin();
}

void loop()
{
   cdino.loop();
}
```

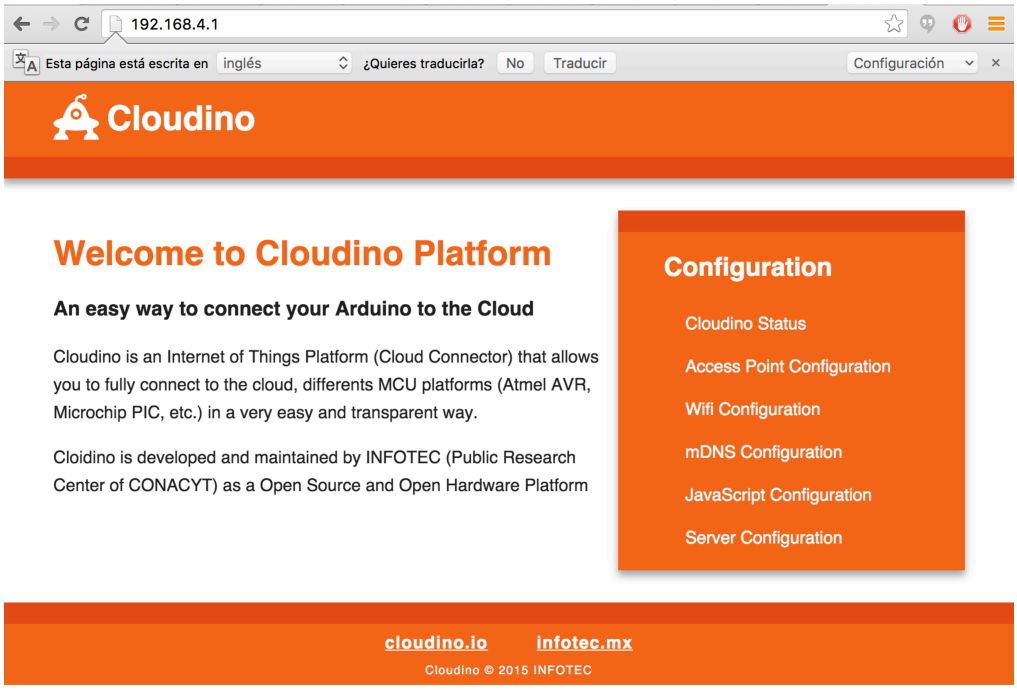
Example of request to FIWARE Context Broker

```
curl https://orion.lab.fi-ware.org:1026/ngsi10/contextEntities/MyHouse -X GET
-s -S --header 'Content-Type: application/json' --header 'Accept:
application/json' --header "X-Auth-Token: kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI" |
python -mjson.tool
```

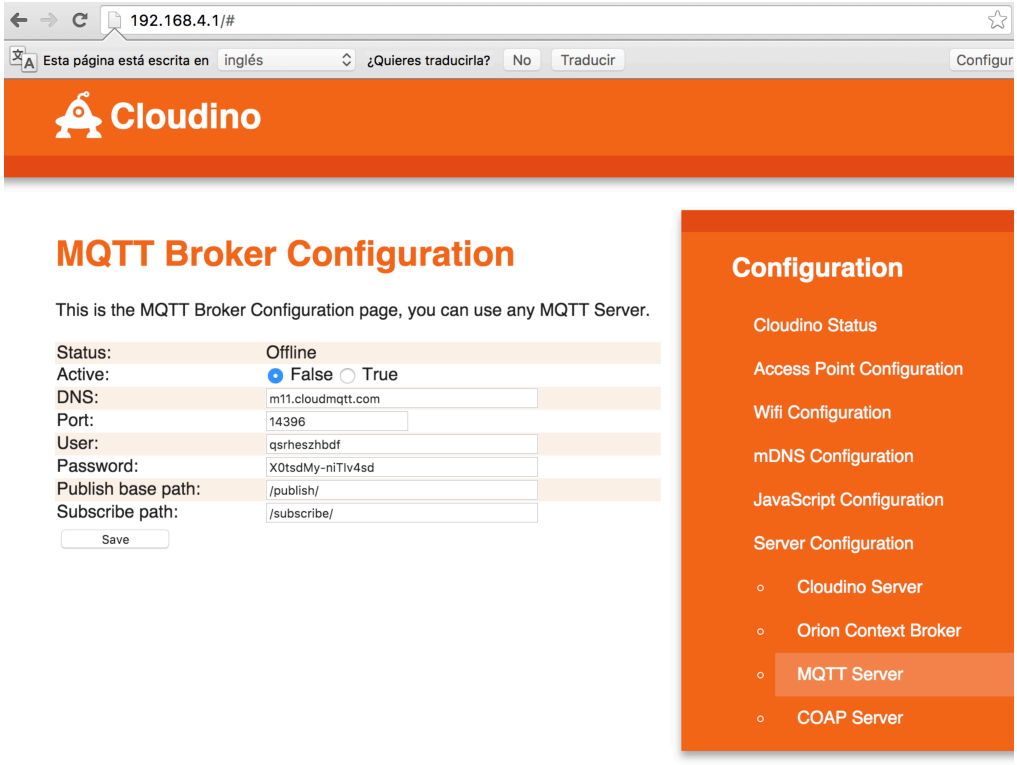
3.2. Configuring Cloudino Connector with MQTT IoT-Agent

Cloudino Connector can connect to the FIWARE using MQTT IoT-Agent, using the simple Cloudino **Configuration Web Interface**.

The **Cloudino Connector** starts an access point that lets you connect to the configuration web interface at: http://192.168.4.1



To configure the MQTT Protocol to connect to FIWARE Context Broker you have to select the MQTT Server in Server Configuration and setting the next fields:



Active: *true*
DNS: [dns or ip to the MTQQ Server]
Port: [port]
User: [user]
Password: [password]
Publish base path: [publish path]
Subscribe path: [subscribe path]

Example of Arduino Code to Post Temperature and Humidity

```
#include <Cloudino.h>
#include <dht11.h>

#define DHT11PIN 8

Cloudino cdino;          //Cloudino Library
dht11 DHT11;             //DHT11 Library

void getSensor()
{
   int chk = DHT11.read(DHT11PIN);
   cdino.post("temperature",String((float)DHT11.temperature,2));
   cdino.post("humidity",String((float)DHT11.humidity,2));
   cdino.print("Timer done!");    //Send to console
}

void setup()
{
   cdino.setInterval(10000,getSensor);  //Timer every 10 seconds
   cdino.begin();
}

void loop()
{
   cdino.loop();
}
```

Example of request to FIWARE Context Broker

```
curl https://orion.lab.fi-ware.org:1026/ngsi10/contextEntities/MyHouse -X GET
-s -S --header 'Content-Type: application/json' --header 'Accept:
application/json' --header "X-Auth-Token: kvx0QWD8CEMd3Yl4B1Ugefnr6WcThI" |
```

```
python -mjson.tool
```

Example of Arduino Code on Receiving Messages

```
#include <Cloudino.h>

Cloudino cdino;

void alarm(String msg)
{
  if(msg=="true")
  {
    digitalWrite(13, HIGH);   //Turn the Alarm ON (HIGH is the voltage level)
  }else
  {
    digitalWrite(13, LOW);    //Turn the Alarm off by making the voltage LOW
  }
}

void setup()
{
  pinMode(13, OUTPUT);
  cdino.on("alarm",alarm);        //receive “alarm” message
  cdino.begin();
}

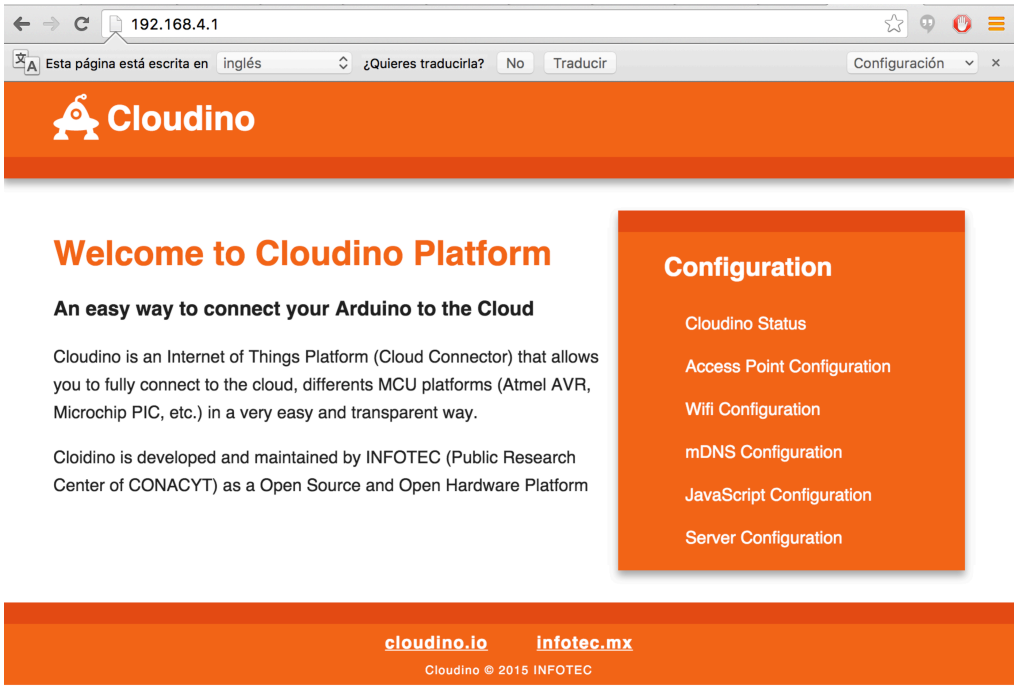
void loop()
{
  cdino.loop();
}
```

3.3. Configuring Cloudino Connector with Cloudino.io Cloud Service (work in process)

Cloudino Connector can connect to the FIWARE using MQTT IoT-Agent, using the simple Cloudino **Configuration Web Interface**.

The **Cloudino Connector** starts an access point that lets you connect to the configuration web interface at: http://192.168.4.1





To configure the Cloudino Connector to connect to Cloudino.io Cloud Service you have to select the Cloudino Server in Server Configuration and setting the next fields:

