



HAL
open science

Présentation d'outils logiciels pour le développement de “ Supervisio ”

Cédric Perrot

► **To cite this version:**

Cédric Perrot. Présentation d'outils logiciels pour le développement de “ Supervisio ”. 16. Journées de la mesure et de la métrologie (J2M), Oct 2021, Ardes-sur-Couze, France. hal-03896016

HAL Id: hal-03896016

<https://hal.inrae.fr/hal-03896016v1>

Submitted on 13 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INRAE



➤ Présentation d'outils logiciels pour le développement de « Supervisio »

PERROT Cédric

Ardes-sur-Couze, 7 Octobre 2021



➤ Présentation de « Supervisio »

➤ Origine de « Supervisio »



Evolution constante des expérimentations au sein de l'unité nécessite :



D'adapter et créer de nouvelles chaînes d'acquisition :

- **Mesures physiques** (T°, humidité, rayonnement, CO2, etc.)
- **Suivi de process** (ex : alarmes chambres de cultures, automatisme Siclex, etc.)

D'assurer la **qualité et la traçabilité des données** horodatées

Création en 2006 et évolution en 2013 d'un **logiciel de monitoring « Supervisio »** sous Labview qui :



- ✓ **Collecte** automatiquement les données à intervalles réguliers
- ✓ **Corrige** les données avec des coefficients d'étalonnage
- ✓ **Stocke** les données dans des fichiers
- ✓ **Émet des alertes** en temps réel
- ✓ Permet aux utilisateurs de **visualiser** sous forme de **graphique** et **d'exporter les données**



- Données stockées dans des fichiers .csv
- 1 seule interface partagée par plusieurs utilisateurs



En 2020, création d'une nouvelle version de « **Supervisio** » orientée **Web**

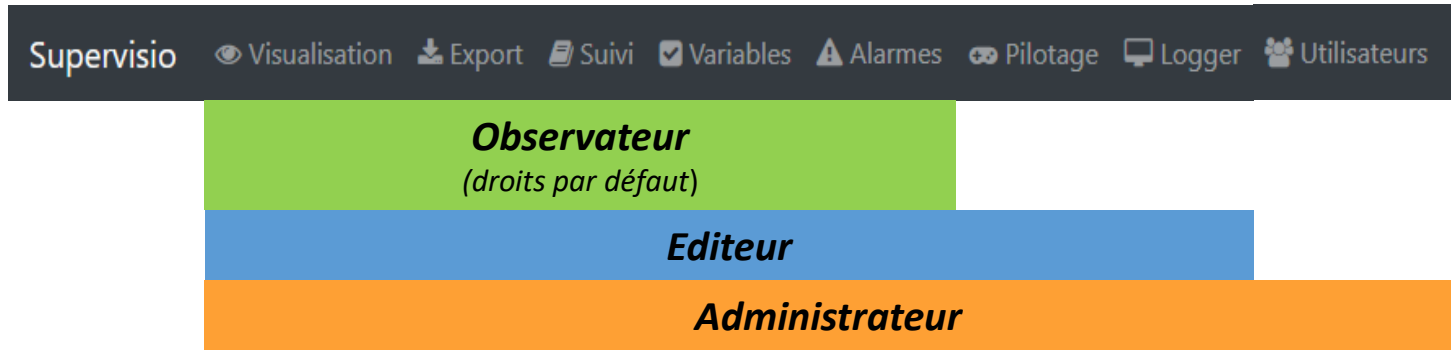
INRAE

Présentation d'outils logiciels pour le développement de « Supervisio »

 Ardes-sur-Couze / 7 Octobre 2021 / Perrot Cédric

- Compte utilisateur

- **accès aux données après identification** depuis n'importe quel navigateur web, dans et hors l'intranet INRAE
- **3 groupes** pour définir les **accès à des fonctionnalités spécifiques** :



Seuls les éditeurs et les administrateurs peuvent créer/modifier des variables

- Stockage dans une base de données de séries temporelles (TSDB **InfluxDB**)
- Exportation optimisée avec possibilité d'agrégation des données (ex : moyenne/jour, cumul/heure)
- Suivi des interventions effectuées sur les expérimentations
- Historique des variables (ex : coefficients d'étalonnage, localisation du capteur)
- Pilotage d'équipements à distance (module à développer)



**Renforce la qualité
et la traçabilité
des données**

Visualiser les données



The screenshot shows the Supervisio web interface. At the top, there's a navigation bar with 'Supervisio' and various menu items like 'Visualisation', 'Export', 'Suivi', 'Variables', 'Pilotage', 'Logger', 'Alarmes', and 'Utilisateurs'. Below this is a header area with the text 'Paramétrez la visualisation de vos données'. The main content area is divided into several sections:

- 1**: A time range selector showing 'Historique' from '19/09/2021 09:27' to '23/09/2021 09:27'. A refresh button is on the right.
- 2**: A 'Localisations' dropdown set to 'barillot/carbhydro/ch1/ble_1' and a 'Filtre' button.
- 3**: An 'Echantillonnage' section with 'Unique (Last)' and 'Auto / 1m' options.
- 4**: A table of variables with columns for 'Paramètres', 'Niveau 1', 'Niveau 2', 'Niveau 3', 'Niveau 4', 'Niveau 5', 'Unités', and 'Variables'. Each column has 'Tout' and 'Aucun' buttons. The 'Variables' column is highlighted in green.
- 5**: A 'Visualiser' button at the bottom right.

- 1. Période** à visualiser : Historique ou temps précédent la date/heure actuelle (ex : 6 heures)
- 2. Favoris** pour appliquer rapidement les filtres de sélection (facultatif)
- 3. Echantillonnage** des données : **Type** (ex : *moyenne, cumul, ...*) - **Intervalle** (ex : *Auto, 1 minute, 1 jour, ...*)
- 4. Filtres** de sélection des variables (*Librairie Javascript* : *DataTables*)
- 5. Visualiser** les courbes des variables sélectionnées

➤ Affichage des courbes



Veillez patienter !

Création des courbes en cours

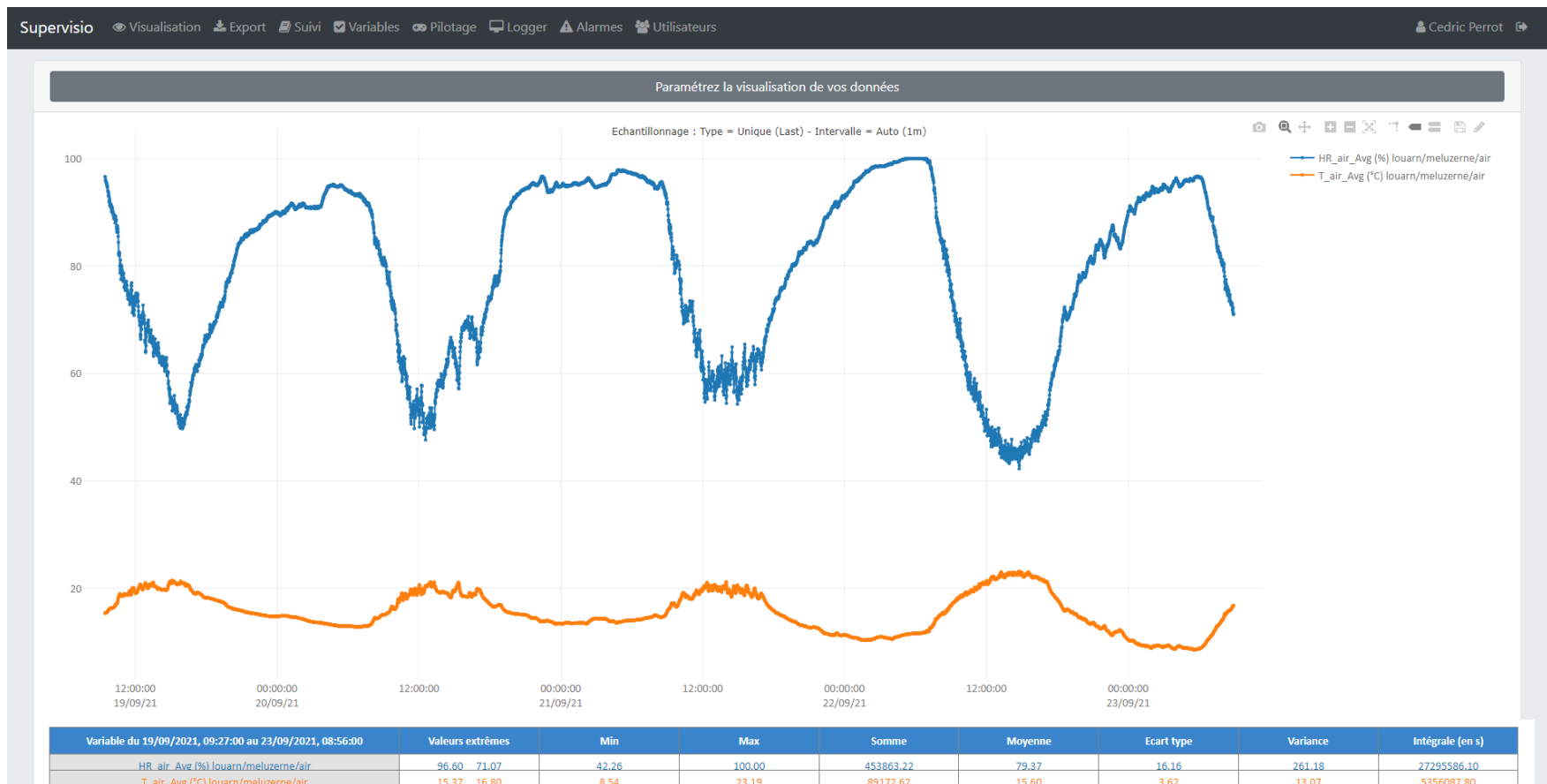


Dates incorrectes!

Veillez modifier les dates et heures pour visualiser votre graphique.

OK

Fenêtres de dialogue (*librairie Javascript* **SweetAlert for Bootstrap**)



➤ Graphique et tableau de calculs mathématiques générés avec **plotly** (*Librairies Javascript, Python, R, Matlab, ...*)



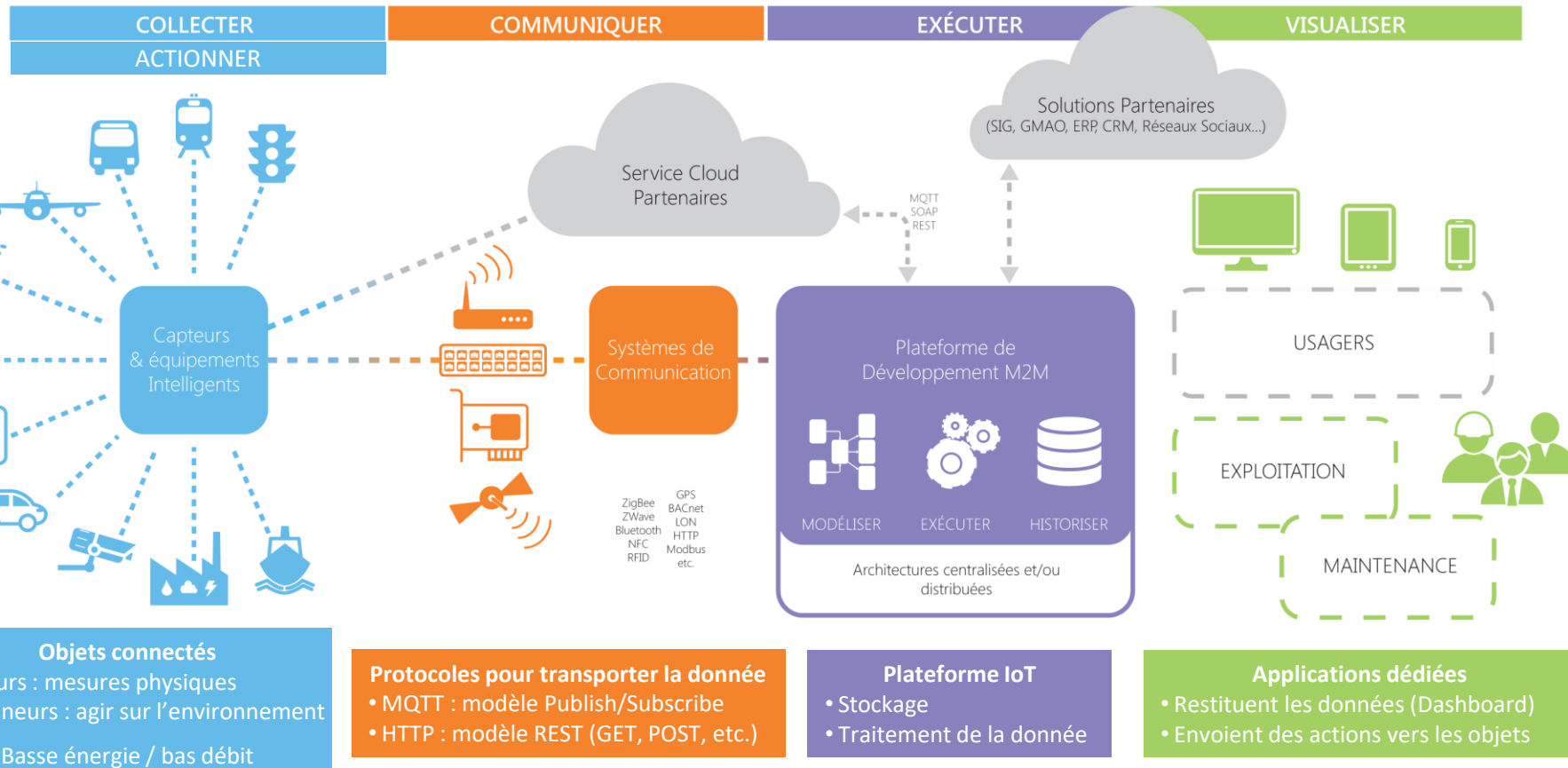
➤ Architecture de « Supervisio »

➤ IoT (Internet des objets)



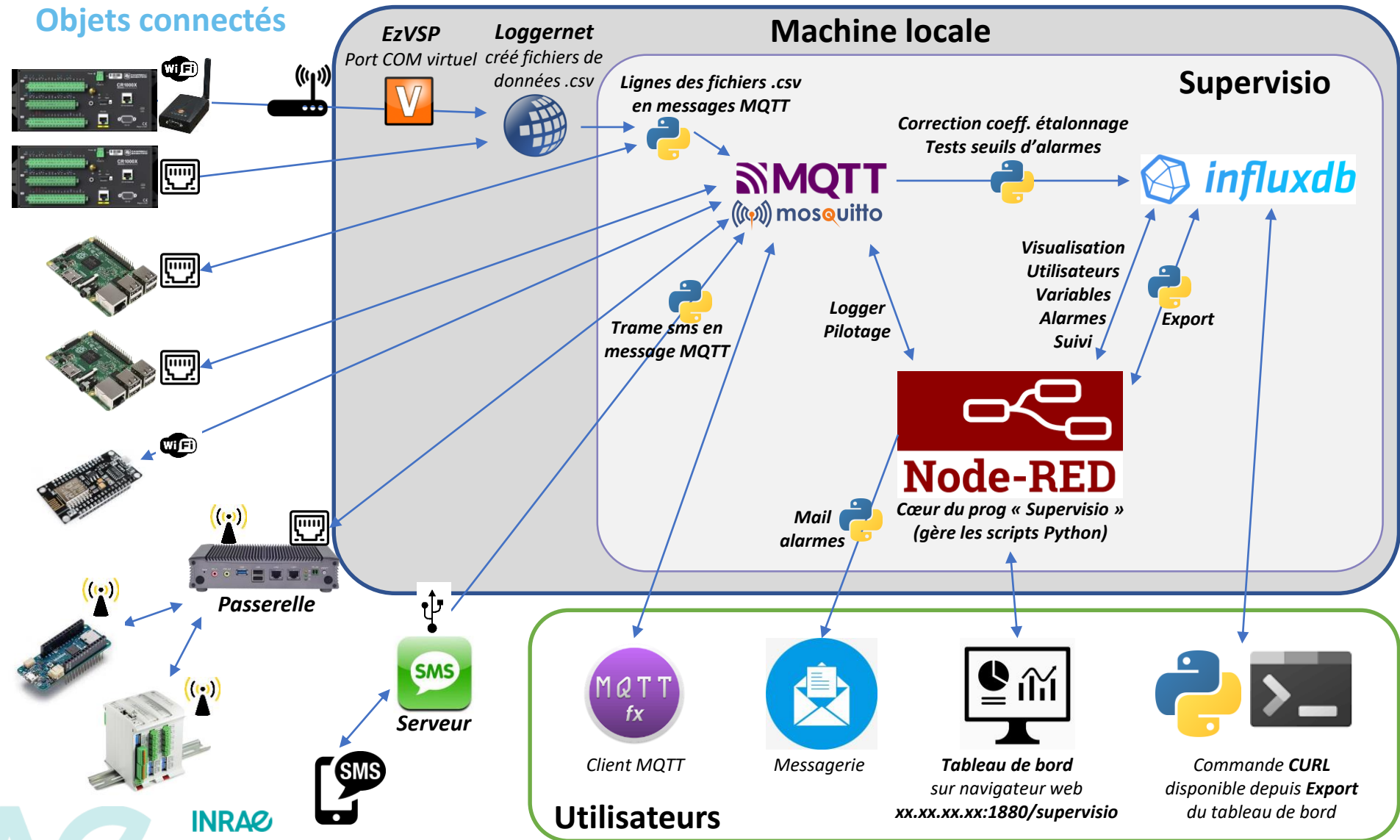
Inventé en 1999

- **Interconnexion de dispositifs informatiques permettant d'envoyer et de recevoir des données via Internet**
- Englobe généralement du **hardware**, des **protocoles de communication**, du **software**, du **cloud** et du **mobile**



source: <https://blog.engineering.publicissapient.fr/2015/12/02/linternet-des-objets-101/>

Objets connectés





➤ Outils logiciels

➤ Pourquoi utiliser Node-RED ?



Node-RED

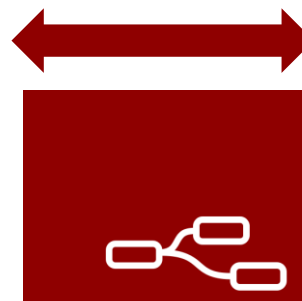
- Outil de programmation visuelle simple, puissant et libre pour construire des applications de l'Internet des Objets

Sources de données

- Ports série
- GPIO microcontrôleurs (ex : Raspberry PI, Arduino)
- Message réseau (ex : mails, MQTT)
- ...

Stockage de données

- Base de données (ex : InfluxDB, MySQL)
- Fichiers
- FTP / SFTP
- ...



Node-RED

automatise les connexions

Et bien d'autres possibilités !



Machine learning

Tableau de bord (utilisateurs)

- Voyants
- Boutons de commande
- Graphiques
- Images
- ...

- Communauté importante

<http://noderedguide.com/>



Forum



Slack team



Stack Overflow



GitHub

INRAE

Présentation d'outils logiciels pour le développement de « Supervisio »



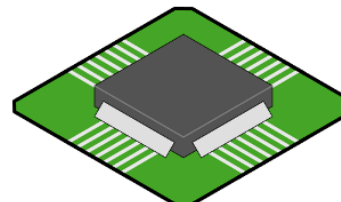
Ardes-sur-Couze / 7 Octobre 2021 / Perrot Cédric

➤ Avant de programmer...

Installation sur Windows, Linux, Android, ...



En local



Sur un matériel, ex :

- Raspberry Pi
- BeagleBone Black



Sur le cloud

Source : <https://nodered.org/>

1. Installer Node.js® (environnement d'exécution Javascript - <https://nodejs.org/en/>)
2. Dans l'éditeur de ligne de commande, renseigner : (*sudo pour linux*) **`npm install -g --unsafe-perm node-red`**
3. Démarrer le serveur Node-RED avec la commande :



4. Pour lancer Node-RED automatiquement :

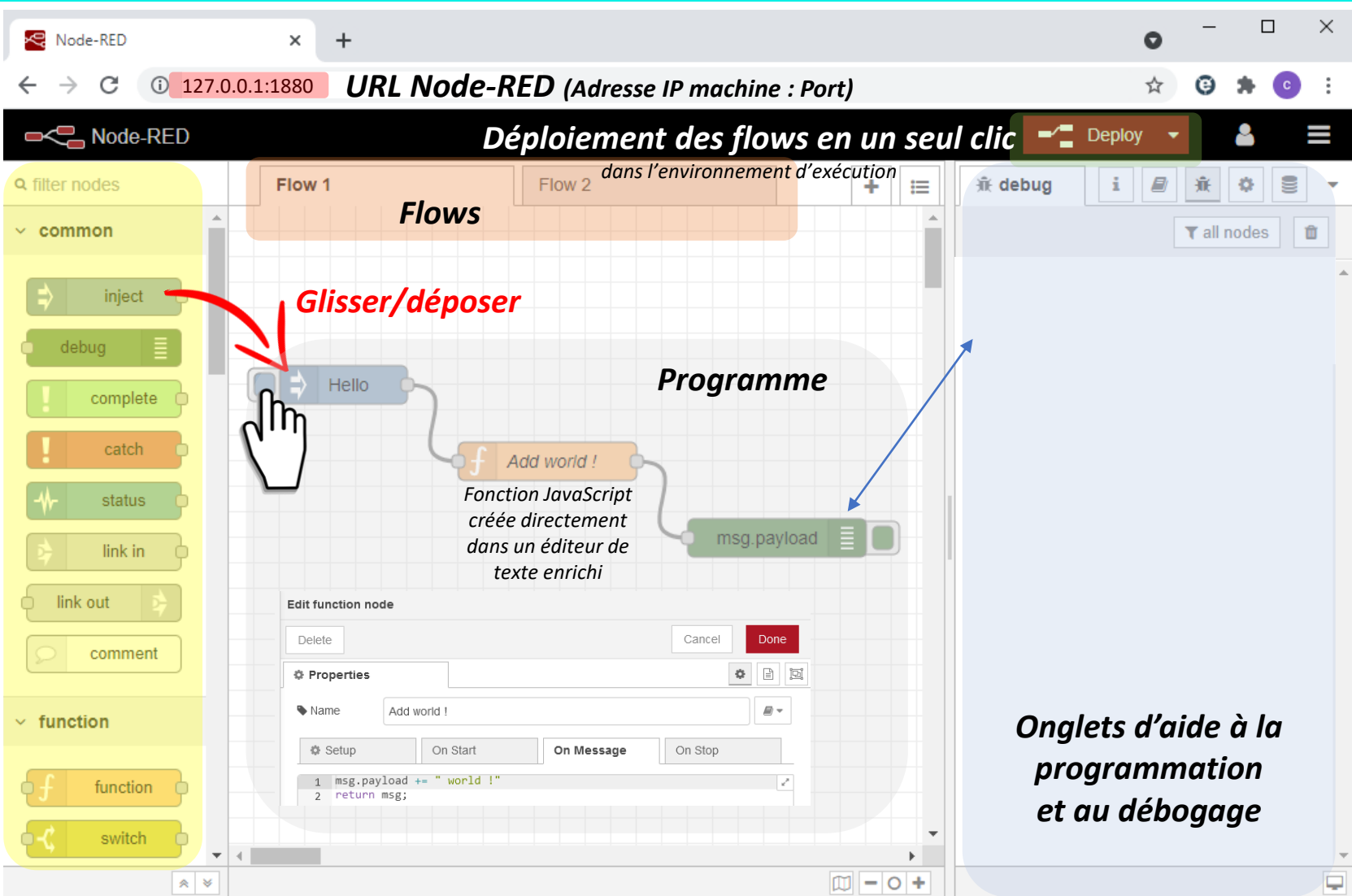


```
pi@raspberrypi:~$ node-red
23 Sep 13:41:42 - [info]
Welcome to Node-RED
=====
23 Sep 13:41:42 - [info] Node-RED version: v1.3.5
23 Sep 13:41:42 - [info] Node.js version: v10.24.0
23 Sep 13:41:42 - [info] Linux 5.10.17-v7+ arm LE
23 Sep 13:41:46 - [info] Loading palette nodes
23 Sep 13:41:50 - [info] Dashboard version 2.30.0 started at /ui
23 Sep 13:41:50 - [info] Settings file : /home/pi/.node-red/settings.js
23 Sep 13:41:50 - [info] HTTP Static : /home/pi/.node-red/static
23 Sep 13:41:50 - [info] Context store : 'default' [module=memory]
23 Sep 13:41:50 - [info] User directory : /home/pi/.node-red
23 Sep 13:41:50 - [warn] Projects disabled : editorTheme.projects.enabled=false
23 Sep 13:41:50 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.json
23 Sep 13:41:50 - [info] Server now running at http://127.0.0.1:1880/
```

URL à renseigner sur votre navigateur web pour accéder à l'interface de Node-RED



➤ Interface de programmation



Palette de Nodes

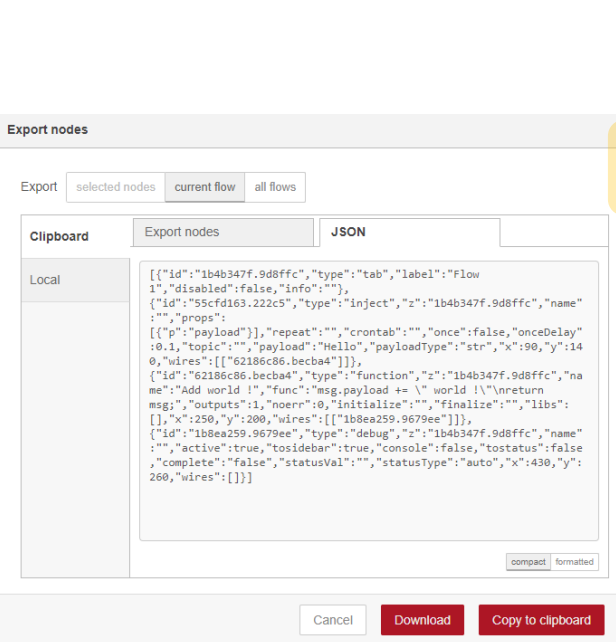
(blocs de code prédéfinis pour effectuer des tâches)

Glisser/déposer

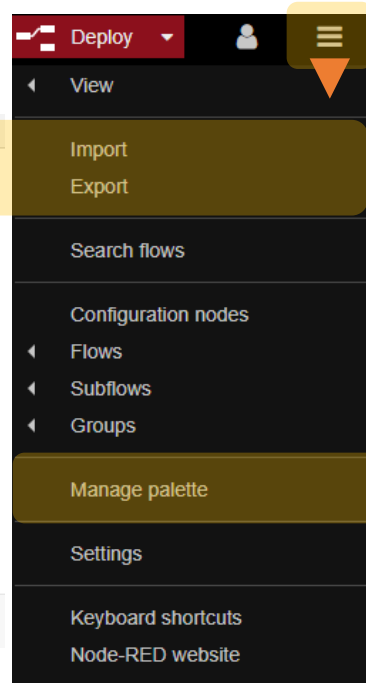
Programme

Fonction JavaScript créée directement dans un éditeur de texte enrichi

Onglets d'aide à la programmation et au débogage



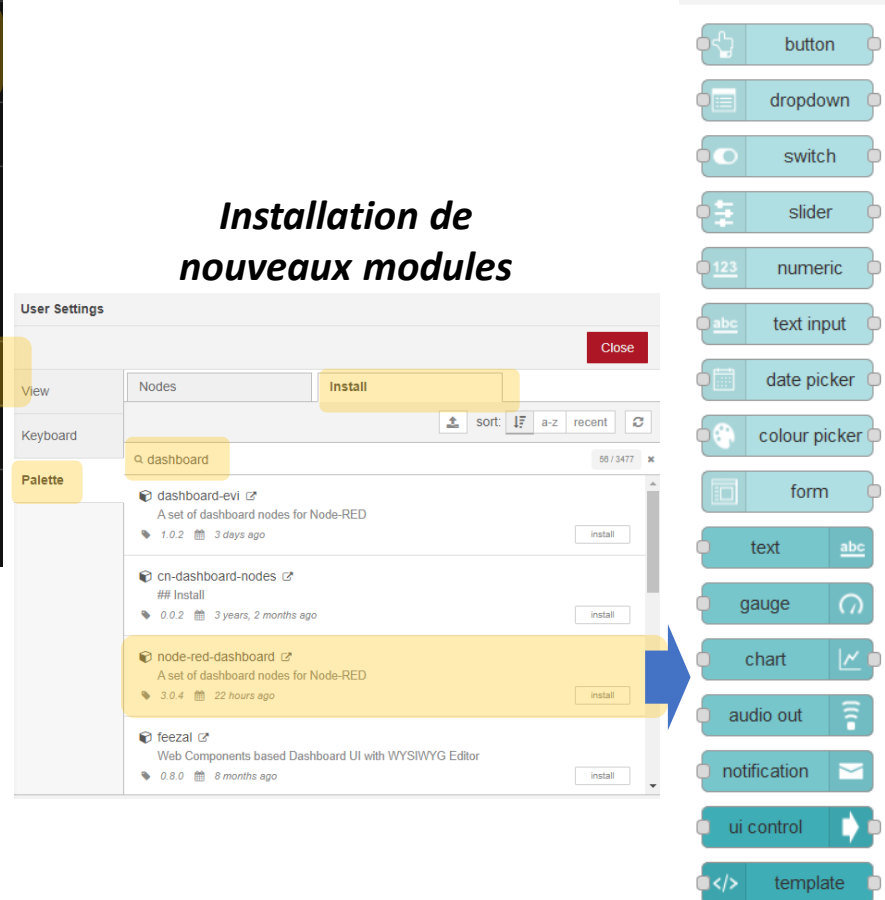
```
[[{"id":"1b4b347f.9d8ffc","type":"tab","label":"Flow 1","disabled":false,"info":""}, {"id":"55cfd163.222c5","type":"inject","z":"1b4b347f.9d8ffc","name":"","props": [{"p":"payload"}]}, {"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"Hello","payloadType":"str","x":90,"y":140,"wires":[["62186c86.becba4"]]}, {"id":"62186c86.becba4","type":"function","z":"1b4b347f.9d8ffc","name":"Add world !","func":"msg.payload += \" world !\\n\\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":250,"y":200,"wires":[["1b8ea259.9679ee"]]}, {"id":"1b8ea259.9679ee","type":"debug","z":"1b4b347f.9d8ffc","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"statusVal":"","statusType":"auto","x":430,"y":260,"wires":[]}]
```



- Deploy
- View
- Import
- Export
- Search flows
- Configuration nodes
- Flows
- Subflows
- Groups
- Manage palette**
- Settings
- Keyboard shortcuts
- Node-RED website

Menu

Ajout à la palette de nodes



Installation de nouveaux modules

node-red-dashboard

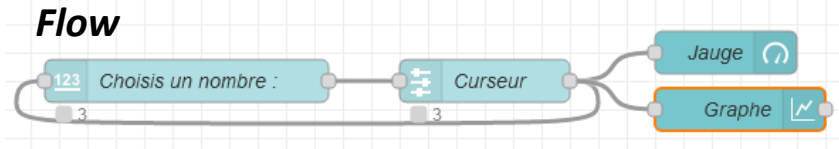
- button
- dropdown
- switch
- slider
- numeric
- text input
- date picker
- colour picker
- form
- text
- gauge
- chart
- audio out
- notification
- ui control
- template

Import / Export des flows au format JSON

- Exemples de flows disponibles sur internet
- Partager des flows avec d'autres
- Sauvegarder le projet



➤ Tableau de bord



Edit chart node

Delete Cancel Done

Properties

Group: [Test] J2M

Size: auto

Label: Graphe

Type: Line chart enlarge points

X-axis: last 1 hours OR 1000 points

X-axis Label: HH:mm:ss as UTC

Y-axis: min max

Legend: None Interpolate: linear

Series Colours: [Color palette]

Blank label: display this text before valid data arrives

Class: Optional CSS class name(s) for widget

Name: Graphe

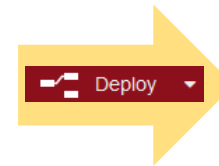
dashboard

Layout Site Theme

Tabs & Links

- Test
 - J2M
 - Choisis un nombre :
 - Curseur
 - Jauge
 - Graphe

Configuration des éléments du tableau de bord



URL du tableau de bord

Nouvel onglet x +

127.0.0.1:1880/ui

Test

J2M

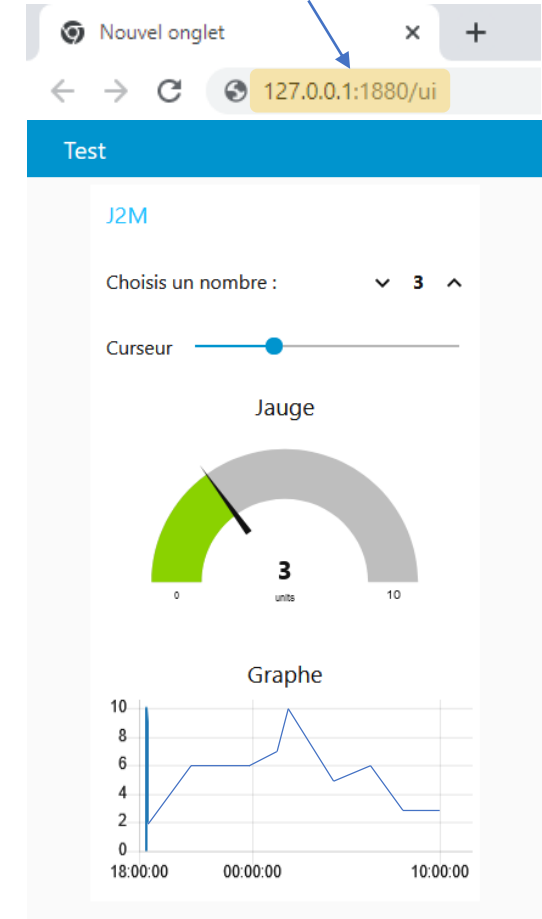
Choisis un nombre : 3

Curseur

Jauge

0 3 units 10

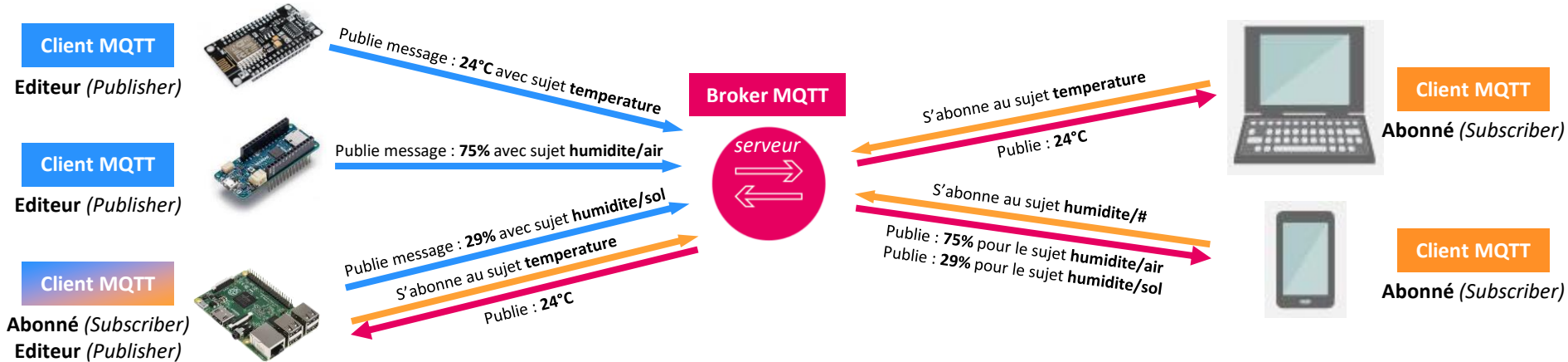
Graphe



- Le module **Dashboard** n'autorise **qu'un utilisateur** à la fois (**mdashboard** pour un fonctionnement **multi-utilisateurs**)
- Pour **Supervisio**, le tableau de bord est créé à partir de **pages HTML** avec des nodes `{ template }`



Protocole de messagerie open source basé sur le TCP/IP
Standard international dans la communication entre les objets IoT

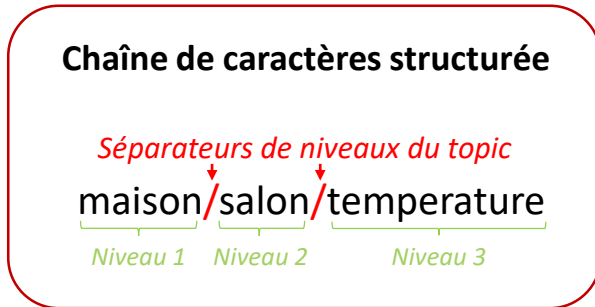


- Un client abonné (*Subscriber*) s'abonne à un ou plusieurs sujets (*Topic*) auprès du serveur (*Broker MQTT*)
- Un client éditeur (*Publisher*) publie un message avec un contenu (*Payload*) et un sujet (*Topic*) vers le serveur
- Le serveur renvoie le message en temps réel uniquement aux clients abonnés au sujet



- ✓ plus rapide et moins énergivore que le *protocole HTTP*
- ✓ simplicité et sa légèreté
- ✓ nécessite une bande passante minimale
- ✓ aucune dépendance entre les clients éditeurs et abonnés
- ✓ facilement extensible puisque le serveur ne gère ni les sujets ni les contenus des messages

- **Topic = sujet du message**



Un client peut s'abonner à plusieurs Topics simultanément

Seulement 1 niveau

maison/+/temperature

- ✓ maison/salon/temperature
- ✓ maison/cuisine/temperature
- ✗ maison/salon/humidite
- ✗ garage/salon/temperature

Multiplés niveaux

maison/salon/#

Seulement à la fin

- ✓ maison/salon/temperature
- ✓ maison/salon/humidite/air
- ✗ maison/cuisine/temperature

- **Payload = contenu du message**

➤ **Chaîne de caractères** d'une **simple valeur** (ex : température) à un **document JSON** (ex : bulletin météo)

- **QoS = qualité de service** : un message toujours publié avec un QoS

| QoS = 0 (message envoyé au plus une fois) | QoS = 1 (message envoyé au moins une fois) | QoS = 2 (message envoyé une seule fois) |
|---|--|--|
| Message est envoyé une seule fois. Aucune garantie de distribution du message | Message envoyé jusqu'à ce que la livraison soit garantie | Message garanti d'être livré 1 seule fois à l'abonné |
| Le message n'est pas stocké | Le message doit être stocké localement | Le message doit être stocké localement |
| Mode de transfert le plus rapide | | Mode de transfert le plus sûr, mais le plus lent |

- **Option Retained** : l'éditeur demande au broker de conserver le contenu du message correspondant au Topic

➤ Un client qui s'abonne au broker récupère immédiatement le dernier message.





Serveurs MQTT :

Plus de 30 brokers disponibles dont un **open source** : Eclipse Mosquitto™







Clients MQTT :

- **Librairies pour presque tous les langages de programmation**

- ✓ Eclipse Paho Python (*pip install paho-mqtt*)  
- ✓ MQTT.js (*npm install mqtt --save*) 
- ✓ PubSubClient (*librairie Arduino*) 

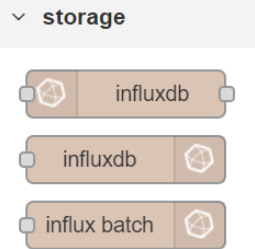
- **Outils et applications :**

- Web 
- Plateformes mobiles Android et iOS  MQTT Dash
- Outils de bureau  MQTT Explorer
- Outils en lignes de commande  MQTT CLI
- Etc.

- ✓ **Base de données de séries temporelles** (TSDB = Time Series Database)
- ✓ **Open source** et qui peut être **utilisé gratuitement** (<https://portal.influxdata.com/downloads/> - InfluxDB 1.x)
- ✓ Rapide pour **stocker** et **traiter** des volumes massifs de **données horodatées**
- ✓ **Monitoring** simples avec **Grafana** ou **Chronograf**
- ✓ Interrogeable via **InfluxQL** (langage de requête de type SQL)



Méthodes d'interrogation et d'insertion de données dans influxDB



Interface en ligne de commande

```
influx.exe - Raccourci
Connected to http://localhost:8086 version 1.8.0
InfluxDB shell version: 1.8.0
> use supervisio
Using database supervisio
> show measurements
```



```
curl -G
'http://localhost:8086/query?pretty=true
--data-urlencode "db=meteo" --data-urlencode "q=SELECT * FROM maison"
```



Installation : `pip install influxdb`
Utilisation : `import influxdb`

```
> USE meteo
Using database meteo
> SELECT * FROM maison LIMIT 10
name: maison
time                capteur hum lieu temp
-----
1572639486660194912 dht22    70.8 cuisine 20.3
1572639546706256357 dht22    70.8 cuisine 20.3
1572639606010049960 dht22    70.9 cuisine 20.3
1572639668443826682 dht22    70.9 cuisine 20.3
1572639726669208565 dht22    70.8 cuisine 20.3
1572639786747649194 dht22    87.1 cuisine 22.1
1572639848684133125 dht22    62.6 cuisine 22.8
```

Database
Requête InfluxQL
Measurement
Keys
Values

Timestamp
(horodatage)

Tags **Fields**

Measurement : groupe de mesures
Tags (étiquettes) : métadonnées indexées pour filtrer, trier, etc.
Fields (champs) : valeurs

```
> SELECT mean(temp) FROM maison WHERE time > now()-6h AND lieu='cuisine' GROUP BY time(1h)
name: maison
time                mean
-----
1572670800000000000 21.299999999999997
1572674400000000000 21.253333333333334
1572678000000000000 21.200000000000024
1572681600000000000 21.200000000000024
1572685200000000000 22.006666666666664
1572688800000000000 22.553333333333333
1572692400000000000 22.680645161290332
```

Ex. d'agrégation de données pour calculer la moyenne sur l'intervalle de temps défini par GROUP BY time(1h)

Source : <https://mars-hack-lab.fr/?p=453>



***Merci de
votre
attention***