INFLUX Base de données InfluxDB

GColin

1 – Présentation de InfluxDB

InfluxDB est une base de données qui enregistre les informations en fonction du temps (*timestamp*). Elle permet notamment de définir une période d'existence des données après laquelle elles sont automatiquement effacées, ce qui n'est pas le cas pour une base de données comme MySQL, plus connue et plus adaptée aux bases de données relationnelles, comme des fiches clients par exemple.

InfluxDB est très utilisée dans le domaine des objets connectés.

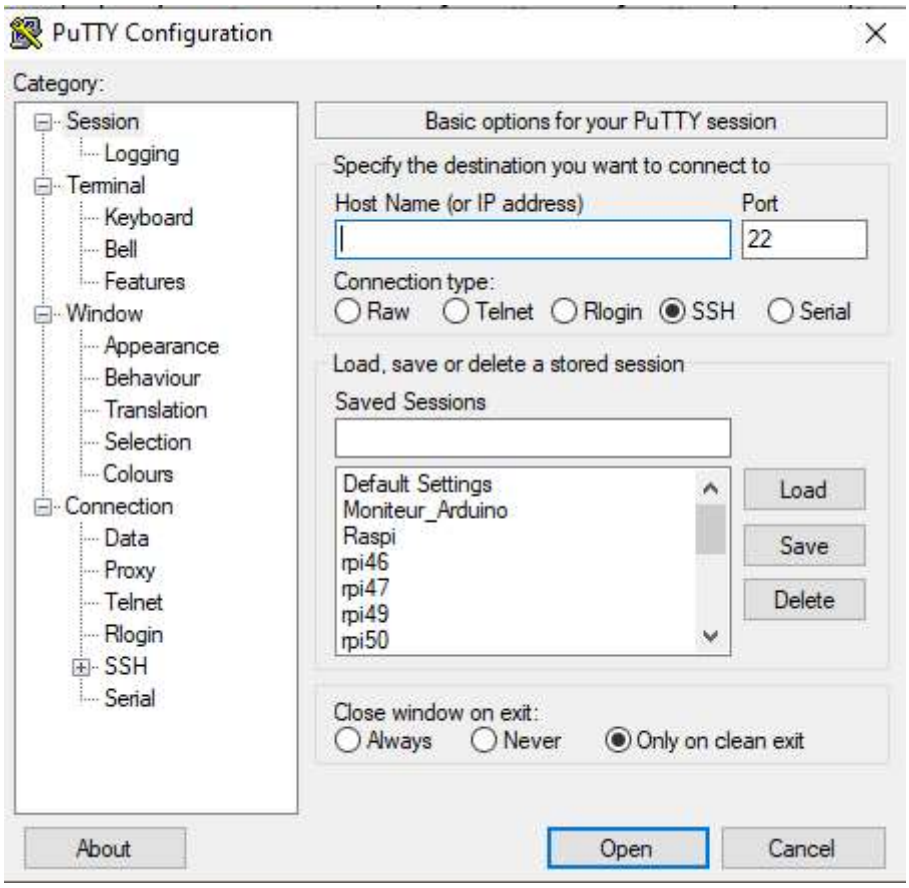
Influxdb utilise 2 langages : InfluxQL et Flux. On se limitera ici à InfluxQL (Query Langage).

2 – Base de données InfluxDB installée sur Raspberry

Dans le cadre des TP, la base de données sera installée sur une Raspberry, qui nous sert de serveur, gérée sous Linux.

L'accès à distance à la Raspberry (et plus généralement les serveurs Linux), peut se faire avec un client SSH, comme le logiciel PUTTY.

L'adresse IP de la Raspberry, le login et le mot de passe seront communiqués par le prof.



⇨ Se connecter à la raspberry hébergeant la base de données.

⇨ Vérifier que InfluxDB est installé sur la Raspberry en tapant influx dans le mode terminal et obtenir l'invite de commande de InfluxDB : (InfluxDB se lance automatiquement au démarrage de la Raspberry).

```
pi@raspberrypi:~ $ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
>
```

3 – Commandes en CLI (ligne de commande)

Pour obtenir l'invite de commande de InfluxDB, taper **influx** :

```
pi@raspberrypi:~ $ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
>
```

Afficher les bases de données créées

> SHOW DATABASES

```
> SHOW DATABASES
name: databases
name
----
_internal
capteur_maison
```

Créer une base de données :

⇨ Si la base de données « capteur_maison » n'est pas créée, créer la.

> CREATE DATABASE capteur_maison

Utiliser une base de données

> USE capteur_maison

Insérer des données dans une nouvelle table

Remarque : la table est appelée ici capteur_salon, mais si plusieurs utilisateurs utilisent la même Raspberry et la même base de données, il est préférable de personnaliser le nom de la nouvelle table en ajoutant les initiales de l'utilisateur par exemple.

Après avoir choisi la base de données (use capteur_maison)

> INSERT capteur_salon,ident=1234 temp=25,humidite=78,co2=450

> INSERT capteur_salon,ident=1234 temp=24,humidite=68,co2=350

Le format est le suivant :

INSERT <measurement>[,<tag-key>=<tag-value>…] <field-key>=<field-value>[,<field2-key>=<field2-value>…] [unix-nano-timestamp]

measurement correspond au nom de la nouvelle table (ici capteur_salon)

tag correspond à une donnée non indexée (exemple une référence, un lieu, …) (ici ident)

field correspond à une donnée indexée (ici temp, humidite et co2)

Un espace sépare les différents champs, une virgule pour les différents tags ou fields.

Afficher les différentes valeurs d'une table

> SELECT * FROM capteur_salon

> SELECT temp,humidite FROM capteur_salon

Afficher la date au format rfc3339 : YYYY-MM-DDTHH:MM:SS.nnnnnnnnnZ

Quitter InfluxDB

> exit

Lancer influxDB avec l'option -precision rfc3339 : influx -precision rfc3339

Visualiser le nouveau format du temps.

Afficher les différentes tables d'une base de données

Après avoir choisi la base de données (use capteur_maison) …

> show measurements

Effacer une table d'une base de données

Après avoir choisi la base de données (use capteur_maison) …

> drop measurement capteur_salon

Autres commandes

Les commandes ci-après sont données à titre indicatif mais ne seront pas mises en œuvre.

Définir une politique de rétention

Cette commande permet de définir la durée de conservation des données, et une duplication sur un autre support.

CREATE RETENTION POLICY <retention_policy_name> ON <database_name> DURATION <duration> REPLICATION <n> [SHARD DURATION <duration>] [DEFAULT]

REPLICATION : duplication sur un autre support (mettre 1 si pas de duplication).

SHARD DURATION : partition des données. Par défaut les données sont mémorisées par groupe de :

DEFAULT : Définit la nouvelle politique de rétention comme politique de rétention par défaut pour la base de données. Facultatif.

Afficher la politique de rétention

> SHOW RETENTION POLICIES ON capteur_maison

```
> SHOW RETENTION POLICIES ON capteur_maison
name    duration shardGroupDuration replicaN default
----    -------- ------------------ -------- -------
autogen 0s       168h0m0s           1        true
un_jour 24h0m0s  1h0m0s             1        false
```

Modifier une politique de rétention

ALTER RETENTION POLICY <retention_policy_name> ON <database_name> DURATION <duration> REPLICATION <n> SHARD DURATION <duration> DEFAULT

Supprimer une politique de rétention

```
DROP RETENTION POLICY <retention_policy_name> ON <database_name>
```

Effacer une base de données

DROP DATABASE "database_name"

Ajouter un utilisateur

```
CREATE USER name_user WITH PASSWORD 'motdepasse'
```

Donner tous les droits à un utilisateur

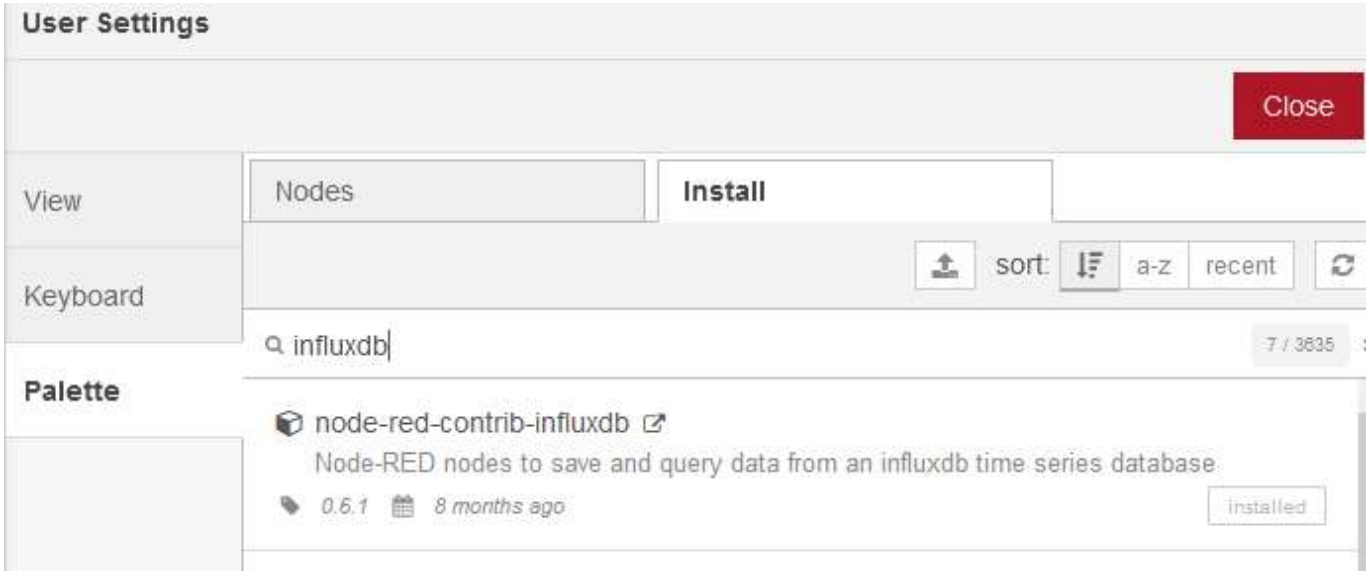
```
GRANT ALL ON base_donnees TO name_user
```

Autres commandes à voir sur Internet… https://docs.influxdata.com/influxdb/v1.8/query_language/

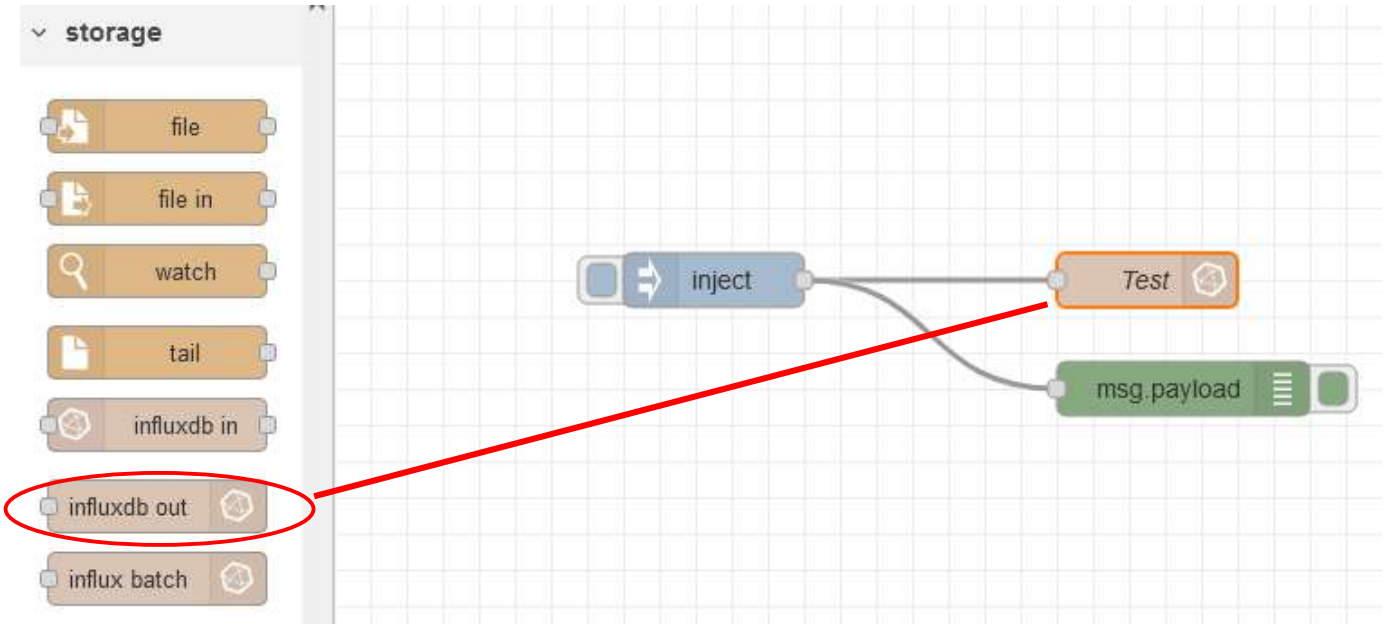
4 – Enregistrer des données avec Node-RED

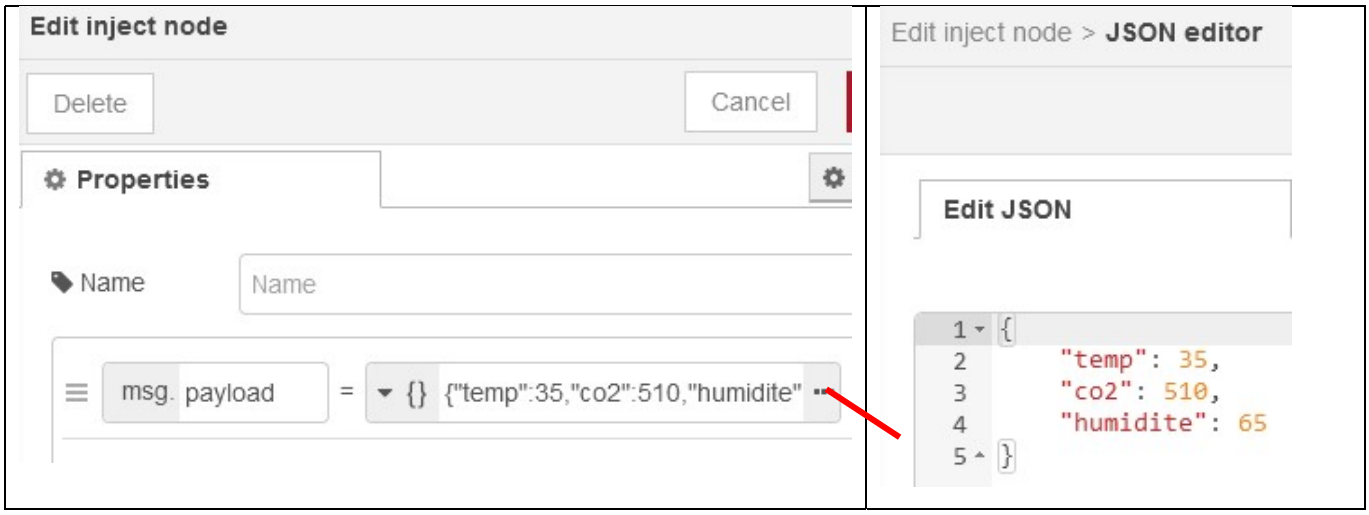
⇨ Sous Node-RED, vérifier la présence des nodes Influxdb (sous menu storage) .

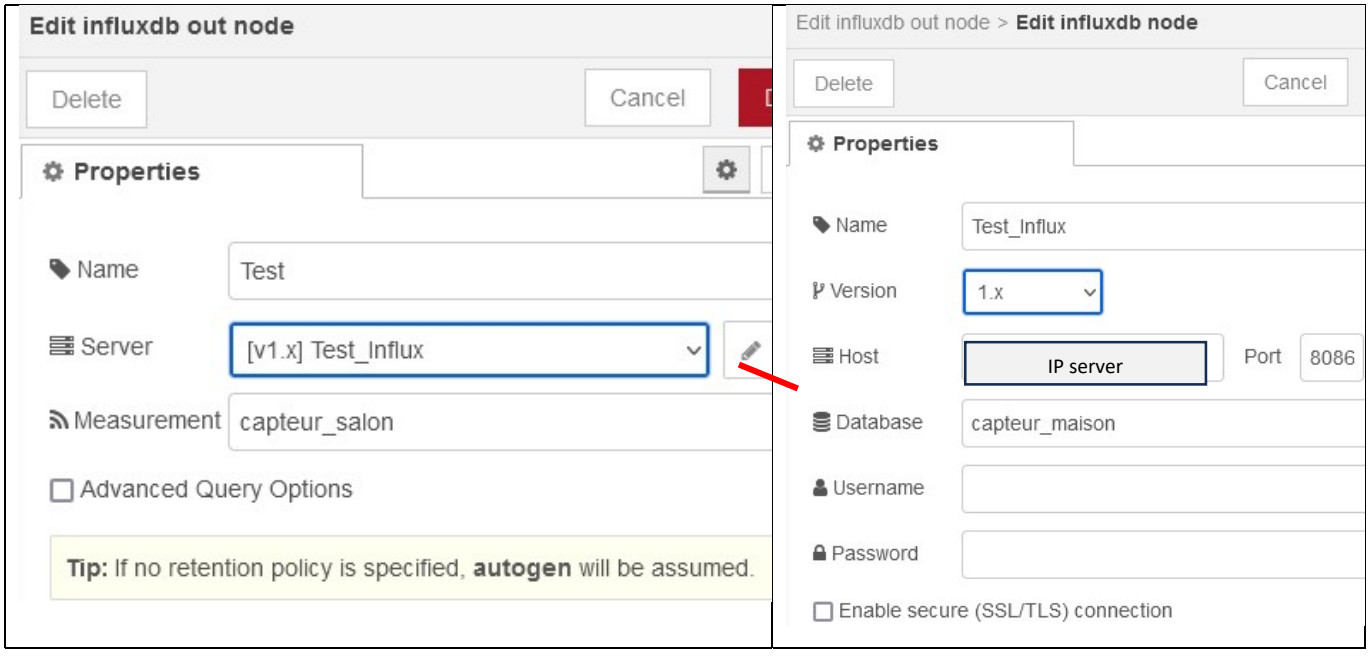
Sinon il faut ajouter node-red-contrib-influxdb, dans le menu Manage palette (attention avec le proxy).



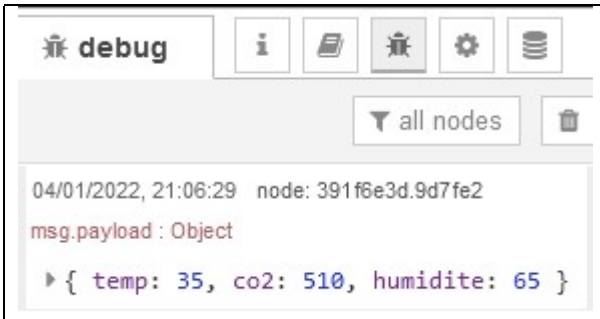
⇨ Réaliser le flow suivant :





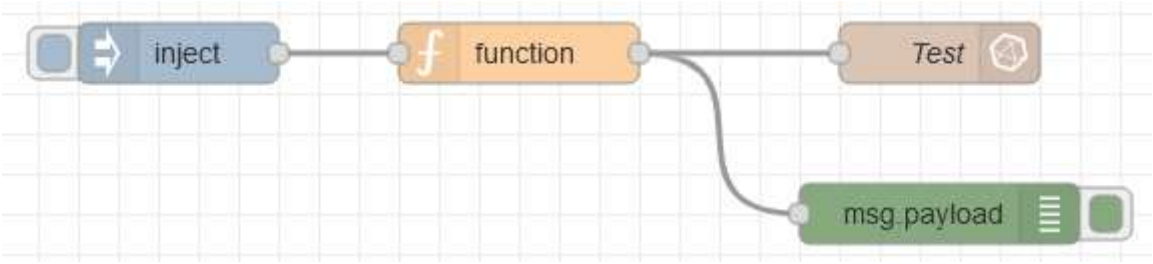


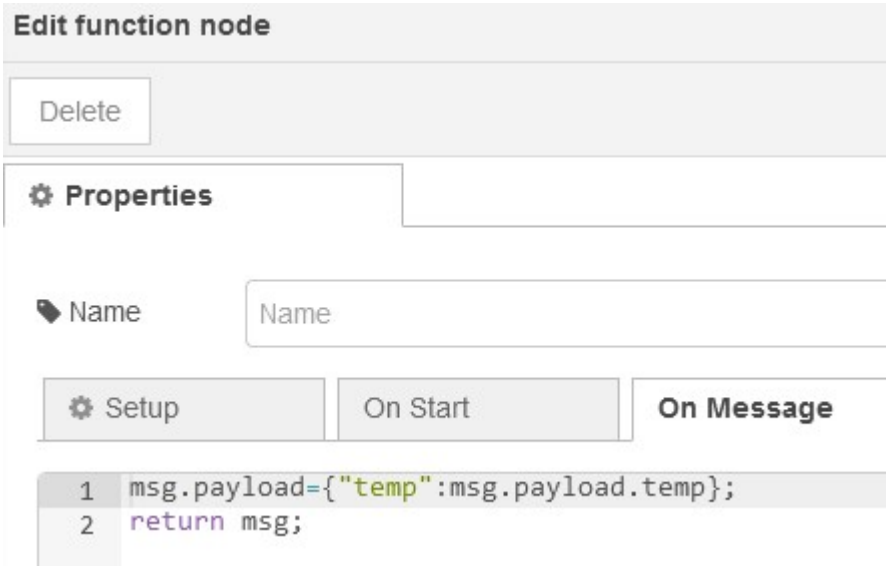
⇨ Injecter des données et observer la fenêtre DEBUG. Vérifier les données enregistrées dans la base de données, avec la commande SELECT dans une fenêtre CLI.




```
> SELECT co2,humidite,temp FROM capteur_salon
name: capteur_salon
time                           co2 humidite temp
----                           --- -------- ----
2022-01-03T16:40:36.661547477Z 450 78       25
2022-01-03T16:44:33.129241314Z 350 68       24
2022-01-03T17:43:05.687883119Z 350 68       24
2022-01-03T20:42:42.332448669Z              28
2022-01-03T20:46:41.203Z                    35
2022-01-03T20:48:38.887Z       510          35
2022-01-03T20:59:39.69Z        510          35
2022-01-04T09:57:46.01Z        510          35
2022-01-04T18:11:49.651Z       510          35
2022-01-04T18:18:23.778Z       510          35
2022-01-04T18:18:56.751Z       510          35
2022-01-04T19:33:10.948Z       510          35
2022-01-04T19:33:42.69Z                     35
2022-01-04T19:55:29.707Z       510 65       35
2022-01-04T20:02:37.822Z       510 65       35
2022-01-04T20:06:29.126Z       510 65       35
```

⇨ Ajouter une fonction pour enregistrer uniquement la température, comme ci-dessous.



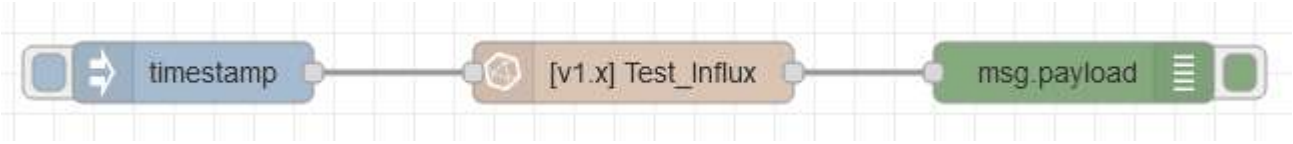


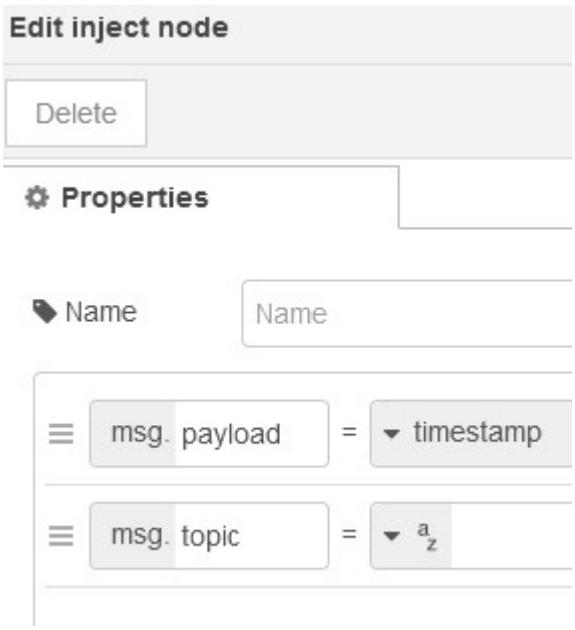
⇨ Vérifier l'inscription dans la base de données.

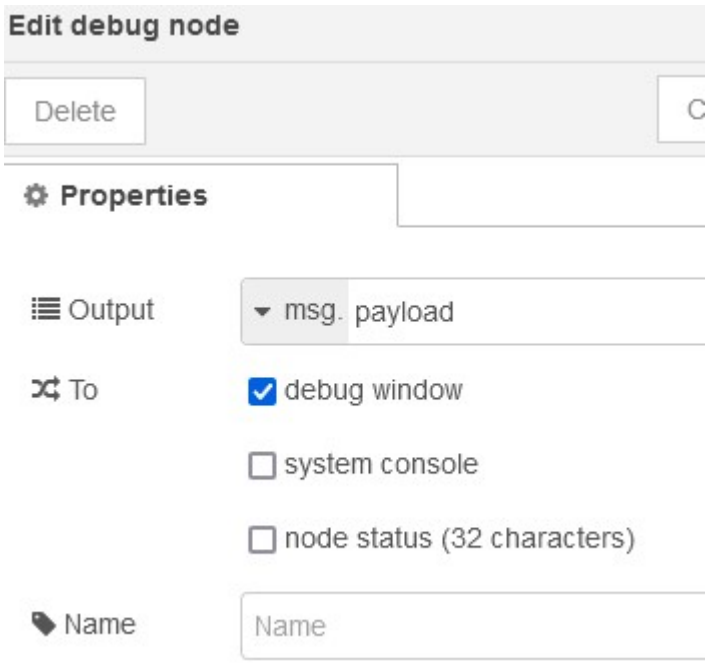
⇨ Modifier la fonction pour enregistrer la température et l'humidité uniquement. Vérifier.

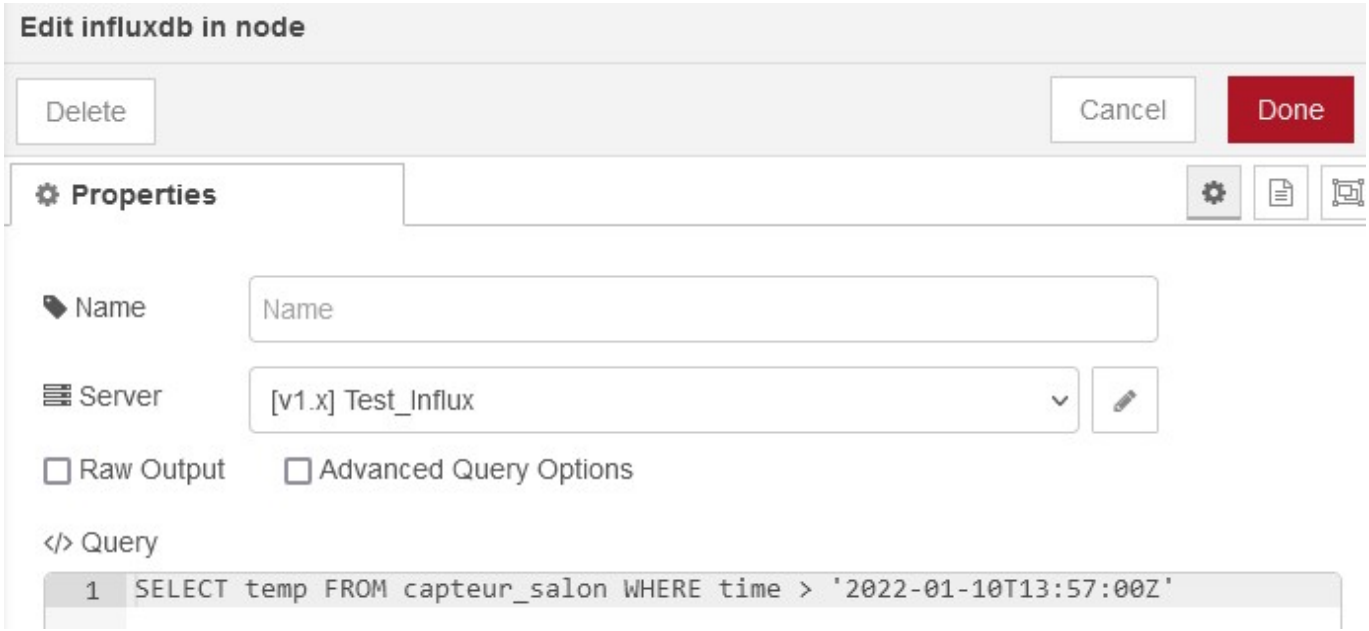
5 – Lire des données dans influxdb avec Node-RED

A la page précédente, ajouter le flow suivant (node : influxdb in)





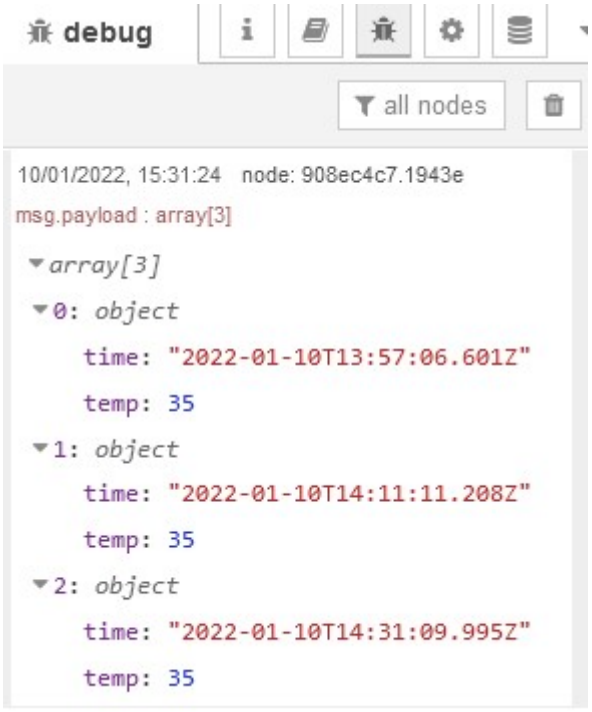




Dans l'édition de Query ci-dessus, seules les mesures enregistrées après le 10/01/2022 à 13h57mn00s (**Attention : heure UTC**) seront sélectionner.

⇨ Adapter la date avec la date courante dans l'édition de Query

⇨ Observer les valeurs lues dans la fenêtre DEBUG comme ci-dessous (cliquer sur l'onglet de timestamp)



⇨ Modifier Query pour afficher l'humidité également.