



Créer un flux de données et une interface utilisateur avec Node-RED



[Mise à jour le 2/7/2022]

• Sources

- Référence - [Guide de l'utilisateur](#) sur le site node-RED.org
- Tutoriel [Installez Node-RED sur Raspberry Pi \(fonctionne sur RPI OS 64 bits\)](#)
- Tutoriel ["Bien débiter avec le Broker MQTT Mosquitto sur Raspberry Pi"](#)
- Tutoriel ["Démarrer avec le tableau de bord Node-RED"](#)

• Lectures connexes

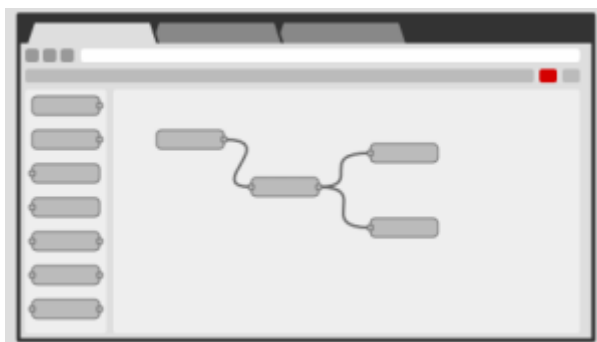
- [Wiki Raspberry Pi sous Linux - Installer un broker MQTT Mosquitto sur un Raspberry Pi](#)
- [Wiki Réseau - Test d'un broker Mosquitto avec MQTTlens](#)
- [Wiki Arduino - Mise en œuvre d'un client MQTT sur un EP8266 \(ESP32\) Feather Huzzah, MKR1010 ou Arduino Uno Wifi 2](#)
- [Wiki Web - Créer un client MQTT \(Websockets\) avec Eclipse Paho](#)
- [Wiki Raspberry Pi sous Linux - Sauvegarder ses données dans une base TSDB \(InfluxDB\)](#)

• Mots-clés

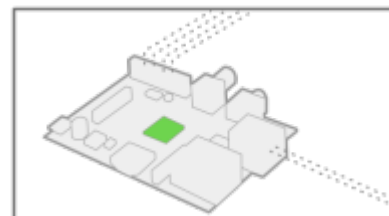
- **node-RED**
 - **node**¹⁾(noeud), **flow**²⁾(flux), **message**³⁾, **payload**⁴⁾(charge utile), **wire**⁵⁾(fil), **workspace**⁶⁾(espace de travail)
- **module dashboard**
 - **dashboard**⁷⁾(tableau de bord), **Layout**⁸⁾(disposition), **group**⁹⁾(groupe), tab¹⁰⁾, **widget**(élément graphique placé dans un groupe)

1. Présentation

Node-RED est un outil de programmation visuel basé sur l'édition de **flux de données**. Il a été développé par IBM pour connecter des périphériques matériels, des API et des services en ligne dans le cadre de l'Internet des objets. L'éditeur de flux est basé sur un navigateur. Ceux-ci sont construits par **glisser-déplacer** et déployés en **un seul clic**.



Le moteur d'exécution est construit sur **Node.js**, tirant ainsi pleinement parti de son **modèle non bloquant**, piloté **par les événements**. Cela le rend idéal pour une utilisation sur du matériel à faible coût, tel que le Raspberry Pi, ainsi que dans le cloud.



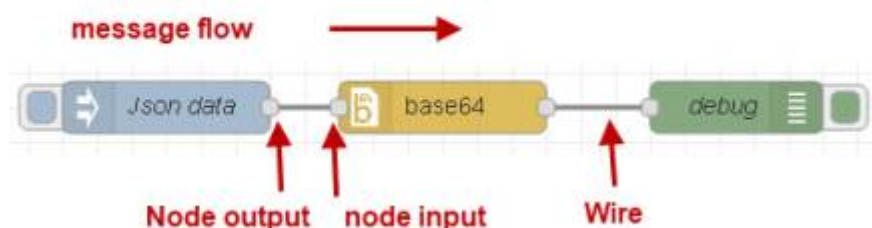
Les flux créés dans Node-RED sont stockés à l'aide de **JSON**, qui peut être facilement **importé** et **exporté** pour être partagé.

2. Concept

- **Source** : [lien](https://nodered.org) sur le site nodered.org.

- **Noeud**

Les noeuds sont les briques de base de Node-Red. Un noeud est une brique logiciel qui traite les messages. Un noeud peut avoir des entrées et des sorties. Il peut accepter des connexions issues de plusieurs noeuds et une sortie peut être transmise à plusieurs noeuds. Chaque nœud a une fonction bien définie et contient ses propres données.



On peut installer des **noeuds supplémentaires** à l'aide du gestionnaire de paquets **npm** ou de l'interface d'administration.

3. Installation de Node-RED et du module dashboard

3.1 Installation de Node-RED

- **Sur un Raspberry Pi**

- **Source** : [Installez Node-RED sur Raspberry Pi \(fonctionne sur RPI OS 64 bits\)](#)



Sur un Raspberry Pi qui tourne sur Raspberry Pi OS (**FULL**), Node-RED est "normalement" déjà installé (sinon, l'installer via *Préférence* → *Add/Remove Software*). Pour le lancer, il suffit de cliquer sur le menu **Programmation** puis **Node-RED**.



Node-RED fonctionnant en tant que **service**, il peut être **lancé** ou **stopper** à partir d'un terminal.

```
node-red-start # démarre le service Node-RED et affiche le journal.  
# Le fait d'appuyer sur ctrl-C ou de fermer la fenêtre n'arrête pas le  
service;  
# il continue à fonctionner en arrière-plan  
node-red-stop # arrête le service Node-RED  
node-red-log # affiche le journal du service
```

- Pour exécuter **Node-RED au démarrage** du Raspberry Pi :

```
sudo systemctl enable nodered.service
```

- Pour **désactiver le service**, exécutez la commande:

```
sudo systemctl disable nodered.service
```



- **Sur un PC sous Windows**

- **Source** : [Running on Windows](#) sur le site nodered.org

1. Installer ou mettre à jour [Node.js](#)
2. Installation de **Node-RED**

Ouvrir une invite de commandes **powershell** ou cmd et vérifier les versions de Node.js et npm, installer node-RED et lancer Node-RED.

*.ps

```
# powershell  
node --version # vérification des versions  
npm --version  
# npm est mis à jour avec l'installation de la nouvelle version de  
Node.js (sur le compte admin)  
npm install -g --unsafe-perm node-red # installation de Node-RED
```

(sur le compte élève)

node-red # *exécution de Node-RED*



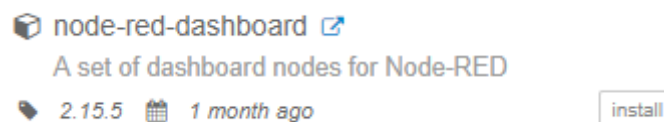
Si Node-RED ne se lance pas avec un message indiquant que le fichier settings.js n'est pas trouvé, créer le fichier ci-dessus dans le répertoire spécifié.



3.2 Installation du module dashboard

- **En mode graphique (à privilégier)**

Ouvrir le menu  et sélectionner **Manage palette** puis l'onglet **install**. Entrer **node-red-dashboard** dans la barre de recherche et l'installer.



- **En mode console**

- **Sur un Raspberry Pi**

Bien que Node-RED soit présent par défaut sous **Raspberry Pi OS**, il est nécessaire d'installer le module Node-RED Dashboard à l'aide des commandes suivantes :

*.bash

```
node-red-stop # Arrêter Node-RED
cd ~/.node-red # se placer dans le répertoire de l'utilisateur
npm install node-red-dashboard # installer le module dashboard
```



Redémarrez ensuite le Raspberry Pi pour vous assurer que toutes les modifications



prennent effet sur le logiciel Node-RED

- **Sur un PC sous Windows**

Ouvrir une invite de commandes cmd ou **powershell**.

*.ps

```
# Powershell
# Si Node-RED est en service, l'arrêter avec ctrl-c.
cd ~/.node-red # se placer dans le répertoire de l'utilisateur
npm install node-red-dashboard # installer le module dashboard
node-red # lancer Node-RED
```

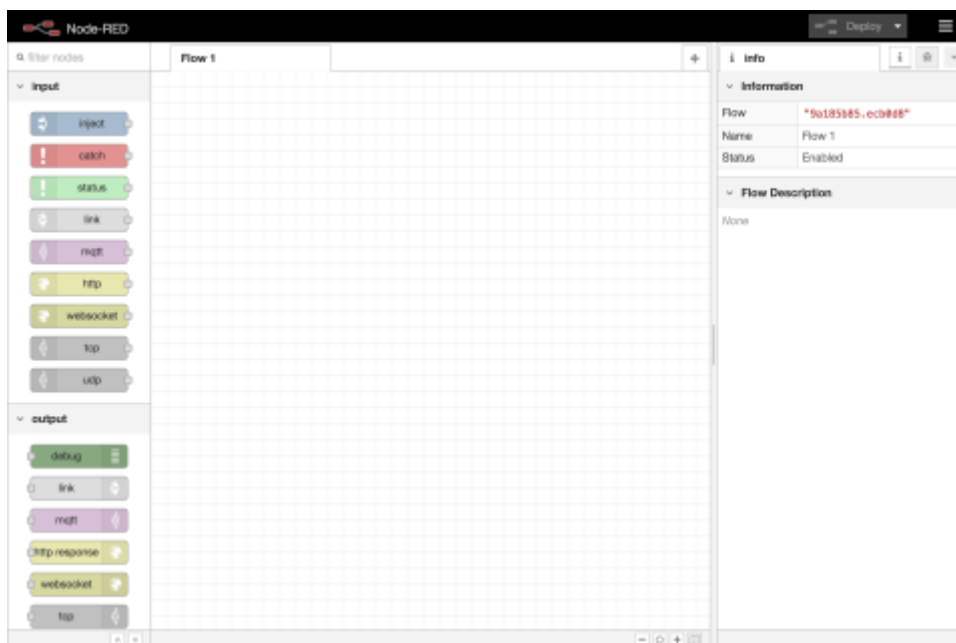


La mise à jour des modules se fait dans l'onglet **Palette** de Manage palette.

3.3 Accès à l'éditeur et au tableau de bord



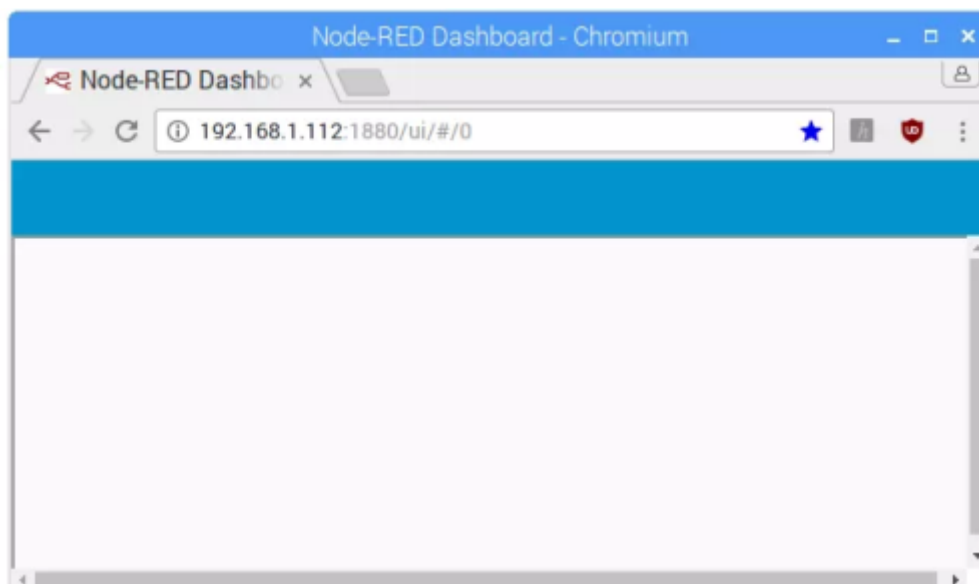
L'accès à l'**environnement de programmation** se fait sur le port **1880** à partir d'un navigateur soit directement sur le Raspberry Pi (localhost: 1880), soit sur un ordinateur (<ip-address>:1880). Il est préférable d'utiliser le navigateur d'un ordinateur connecté sur le même réseau que la Raspberry Pi.



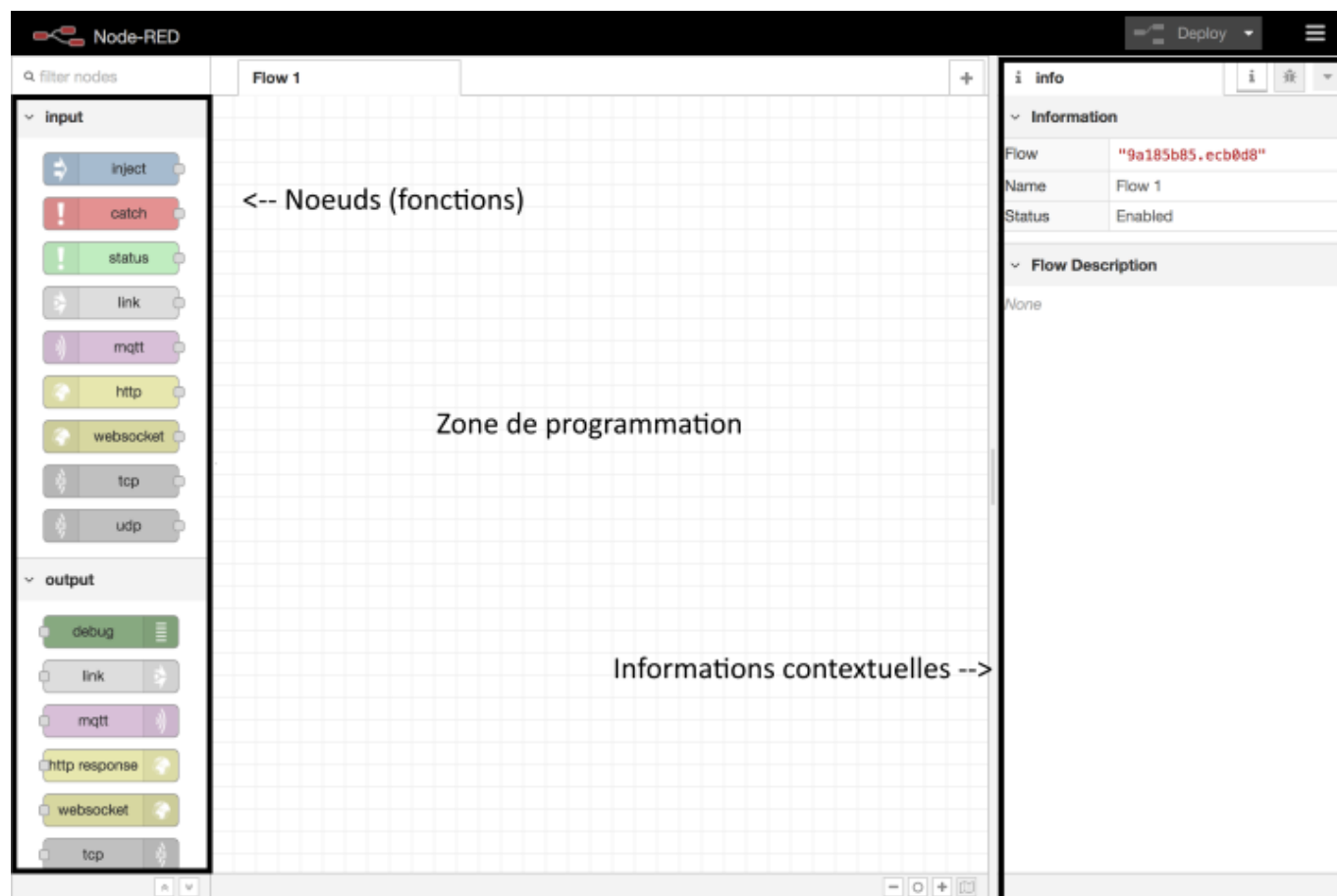
Pour ouvrir l'interface utilisateur Node-RED, tapez l'adresse IP du Raspberry Pi dans un navigateur, suivie de : **1880 / ui**.

Exemple : 192.168.200.42:1880/ui

Pour le moment, le **tableau de bord** est vide comme vous pouvez le voir sur la figure suivante - car nous n'avons encore rien ajouté.

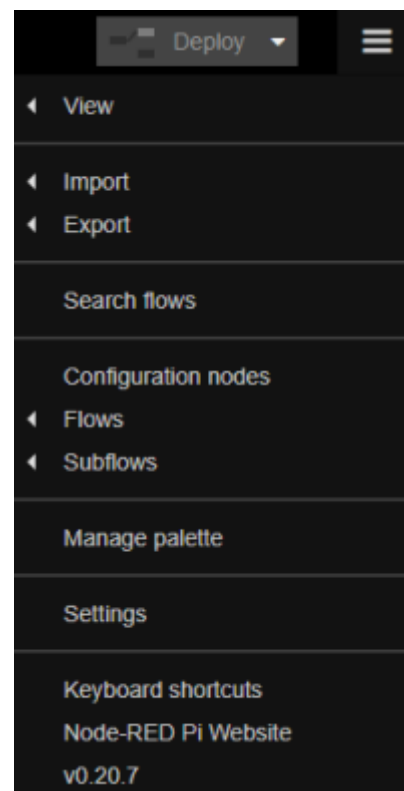


4. Description de l'éditeur



4.1 Le menu

Le menu situé à côté du bouton Deploy contient quelques outils :

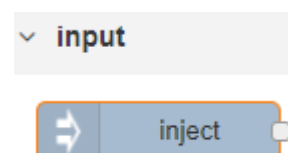


- **View** : Permet d'afficher/masquer la grille qui peut aussi être magnétique (snap to grid)
- **Import** : Permet d'importer des noeuds. C'est avec cette option que vous pouvez ajouter à votre flux des noeuds récupérés sur internet ou sur <http://flows.nodered.org/>
- **Export** : Vous pouvez exporter des nodes vers le presse papier (une chaine de texte au format JSON) ou dans un fichier. Vous trouverez ce fichier dans le répertoire `./home/pi/.node-red/lib/flows`
- **Flows** : Vous pouvez **ajouter** un nouveau flow (Add), **renommer** (Rename), **supprimer** (Delete) le flow en cours

A compléter

5. Créer un premier flux avec Node-RED

- **Objectif** : afficher la date et l'heure dans la console de debogage.
- **Prérequis** - Node-RED est démarré sur un Raspberry Pi accessible. Par exemple, à l'adresse 192.168.200.42
- **a. Accéder à l'éditeur**
Pour **accéder à l'éditeur**, entrez 192.168.200.42:1880 dans un navigateur.

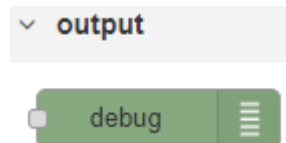


- **b. Ajouter un nœud “Inject”**

Source : [Les nœuds de base sur Node-Red.org](#)

Le nœud Inject vous permet d'injecter des messages dans un flux, en cliquant sur le bouton du nœud ou en définissant un intervalle de temps entre les injections.

Sélectionnez un nœud Inject dans la palette et faites le glisser sur l' espace de travail.



- **c. Ajouter un nœud “Debug”**

Source : [Les nœuds de base sur Node-Red.org](#)

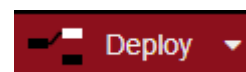
Le nœud de débogage provoque l'affichage de tout message dans la barre latérale *Débogage*



Par défaut, il affiche simplement la charge utile du message (payload), mais il est possible d'afficher l'objet de message entier.

- **d. Relier les nœuds**

Connectez les nœuds Inject et Debug ensemble en faisant glisser le port de sortie de l'un vers le port d'entrée de l'autre.



- **e. Déployer**

À ce stade, les nœuds n'existent que dans l'éditeur et doivent être déployés sur le serveur.

- Cliquez sur le bouton Deploy.

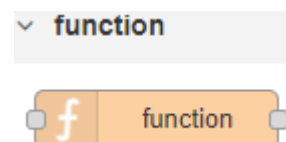
- Sélectionnez l'onglet Débogage  de la barre latérale.

- Cliquez sur le bouton du nœud Inject.

Vous devriez voir des chiffres apparaître dans la barre latérale. Par défaut, le nœud Inject utilise le nombre de millisecondes depuis le 1er janvier 1970 comme charge utile.

Exemple de comportement attendu :

```
03/08/2019 à 11:57:46 node: 26d98d39.dd56f2
msg.payload : number
1564826266571
```

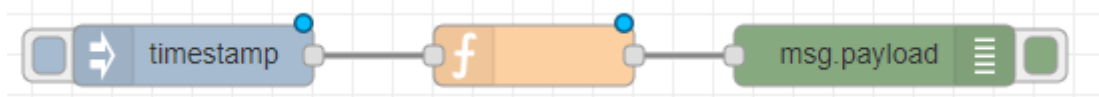


- **f. Ajouter un nœud "Fonction"**

Source : [Fonctions sur Node-Red.org](https://nodered.org/docs/concepts/functions)

Le nœud Fonction vous permet de transmettre un message par l'intermédiaire d'une fonction **JavaScript**.

- **Supprimez** le fil existant (sélectionnez-le et appuyez sur la touche Suppr du clavier).
- **Câblez** un nœud de fonction entre les nœuds d'injection et de débogage.



- **Double-cliquez** sur le nœud Fonction pour afficher la boîte de dialogue d'édition.
- **Copiez** le code suivant dans le champ de fonction:

*.js

```
// Create a Date object from the payload
var date = new Date(msg.payload);
// Change the payload to be a formatted Date string
msg.payload = date.toString();
// Return the message so it can be sent on
return msg;
```

- **Cliquez** sur le bouton de déploiement **Done** puis sur **Deploy**. Désormais, lorsque vous cliquez sur le bouton Injecter, les messages de la barre latérale sont formatés et affichent des horodatages lisibles.

Exemple de comportement attendu :

```
03/08/2019 à 12:18:41 node: 26d98d39.dd56f2
msg.payload : string[65]
"Sat Aug 03 2019 12:18:41 GMT+0200 (heure d'été d'Europe centrale)"
```

- **Résumé**

Cet exemple illustre le concept de base de la création d'un flux. Il montre comment le nœud Inject peut être utilisé pour déclencher manuellement un flux et comment le nœud Debug affiche les messages dans la barre latérale. Il montre également comment le nœud Fonction peut être utilisé pour écrire du code JavaScript personnalisé à exécuter sur des messages.

- **Source**

Le flux créé dans ce tutoriel est représenté par le JSON suivant. Pour l'importer dans l'éditeur, copiez-le dans le presse-papiers, puis collez-le dans la boîte de dialogue Importer.

*.js

```
[{"id":"58ffae9d.a7005","type":"debug","name":"","active":true,"complete":false,"x":640,"y":200,"wires":[]},{id:"17626462.e89d9c","type":"inject","name":"","topic":"","payload":"","repeat":"","once":false,"x":240,"y":200,"wires":[["2921667d.d6de9a"]]}, {"id":"2921667d.d6de9a","type":"function","name":"Format timestamp","func":"// Create a Date object from the payload\nvar date = new Date(msg.payload);\n// Change the payload to be a formatted Date string\nmsg.payload = date.toString();\n// Return the message so it can be sent on\nreturn msg;","outputs":1,"x":440,"y":200,"wires":[["58ffae9d.a7005"]]}]
```

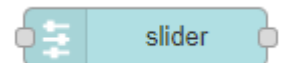
6. Créer un deuxième flux avec Node-RED

- **Objectif** : alimenter une jauge (gauge) avec la donnée issue d'un curseur (slider). Ce flux sera repris au paragraphe suivant lors de la conception d'une interface utilisateur.
- **Prérequis** - Node-RED est démarré sur un Raspberry Pi accessible à l'adresse : @Raspi



a. Accéder à l'éditeur

Pour **accéder à l'éditeur**, entrez @Raspi:1880 dans un navigateur. Nommez l'espace de travail Test (double-clic pour accéder aux paramètres de la page).

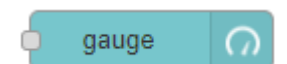


b. Ajouter un noeud "slider"

Source : [node-red-dashboard](#) sur Node-Red.org.

Le nœud slider est un curseur produisant une valeur située dans un intervalle i et s'incrémentant d'un pas p. i et p sont réglables.

Sélectionnez un nœud slider dans la palette et faites le glisser sur l'espace de travail.



c. Ajouter un noeud "gauge"

Source : [node-red-dashboard](#) sur Node-Red.org.

Le nœud gauge se décline en quatre présentations : **standard** (jauge simple), **donut** (rotation à 360°), **compass** (boussole) et **wave** (vague). Vous pouvez spécifier la gamme de couleurs des jauges standard et donut.

Sélectionnez un nœud gauge dans la palette et faites le glisser sur l'espace de travail.

d. Relier les nœuds

Connectez les nœuds *slider* et *gauge* ensemble en faisant glisser le port de sortie de l'un vers le port d'entrée de l'autre.



L'interface utilisateur est conçue dans le paragraphe suivant.

7. Créer une interface utilisateur

- **Source** : [Le tableau de bord Node-RED](https://node-red.org) node-Red.org

7.1 Présentation

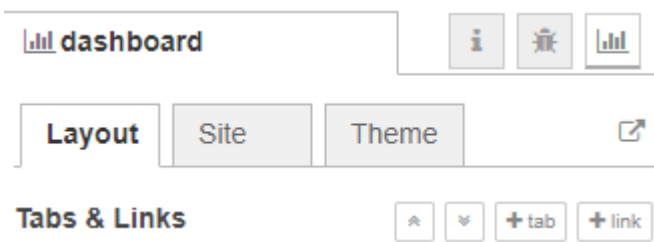
- **a. Organisation des éléments dans une page**
 - Le placement des éléments dans une interface utilisateur se fait sur une **grille**.
 - Chaque élément de **groupe**(group) a une largeur - par défaut, 6 «unités» (une unité mesure 48 pixels de large par défaut avec un intervalle de 6 pixels).
 - Chaque **widget** du groupe a également une largeur - par défaut, "auto", ce qui signifie qu'il remplira la largeur du groupe dans lequel il se trouve, mais vous pouvez le définir sur un nombre fixe d'unités.

A illustrer


- L'algorithme de présentation du tableau de bord essaie toujours de placer les éléments aussi haut et à gauche que possible dans leur conteneur. Cela s'applique à la façon dont les groupes sont positionnés sur la page, ainsi que la façon dont les widgets sont positionnés dans un groupe.
- Pour un groupe de largeur 6, si vous ajoutez six widgets, chacun avec une largeur de 2, ils seront disposés en deux rangées - trois widgets dans chaque.
- Si vous ajoutez deux groupes de largeur 6, tant que la fenêtre de votre navigateur est suffisamment large, ils seront positionnés l'un à côté de l'autre. Si vous réduisez le navigateur, le deuxième groupe se placera en dessous.



Il est conseillé d'utiliser plusieurs groupes plutôt qu'un grand groupe afin que la page puisse se redimensionner de manière dynamique sur des écrans plus petits.



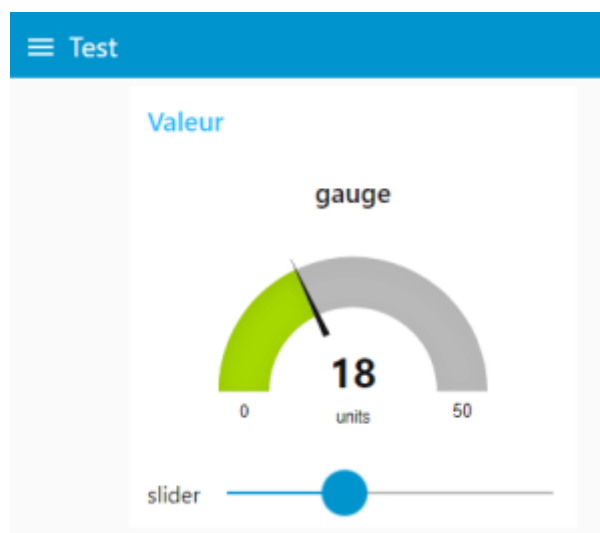
- **b. Mise en page**

- La mise en page de l'interface utilisateur est gérée dans l'onglet **dashboard**  de la barre latérale.

- **Onglet Layout**
 - **+tab** : ajoute des pages à l'interface utilisateur.
 - **+link** : ajoute des liens vers d'autres pages Web.
- **Onglet Site**
 - **Title** : définit le titre de la page.
 - **Options** : gestion de la barre de titre, du menu etc.
 - **Date format** : format de la date pour les graphiques.
 - **Size** : définit la disposition de la grille en pixels. La largeur et la hauteur des widgets peuvent être définies, de même que la largeur des groupes. Ce sont les définitions de base des "unités" utilisées ailleurs dans le tableau de bord.
- **Onglet Theme**
 - **Style** : Vous pouvez sélectionner un thème clair, sombre ou personnalisé. Vous ne pouvez pas avoir différents thèmes pour chaque onglet.
 - **Base Settings** : thème et police de l'interface utilisateur.

7.2 Création de l'interface du deuxième flux

- **Objectif** : afficher le curseur(slider) et la jauge(gauge) dans une interface utilisateur .



- **Prérequis** - Node-Red est démarré sur un Raspberry Pi accessible à l'adresse 192.168.200.42. Le module Dashboard a été correctement installé. Le flux du paragraphe précédent a été créé. Si ce n'est pas le cas importer le code ci-dessous comme cela est expliqué dans la création du premier flux.

*.js

```
[{"id":"7ea54cd1.ce8a34","type":"ui_slider","z":"f2e4faa9.cd1dd8","name":"","label":"slider","tooltip":"","group":"ee35309d.13af6","order":0,"width":0,"height":0,"passthru":true,"outs":"all","topic":"","min":0,"max":50,"step":1,"x":260,"y":280,"wires":[["bc97d2a5.b93b1"]]}, {"id":"bc97d2a5.b93b1","type":"ui_gauge","z":"f2e4faa9.cd1dd8","name":"","group":"ee35309d.13af6","order":1,"width":0,"height":0,"gtype":"gage","title":"gauge","label":"units","format":"{{value}}","min":0,"max":50,"colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":470,"y":280,"wires":[]}, {"id":"ee35309d.13af6","type":"ui_group","z":"","name":"Va
```

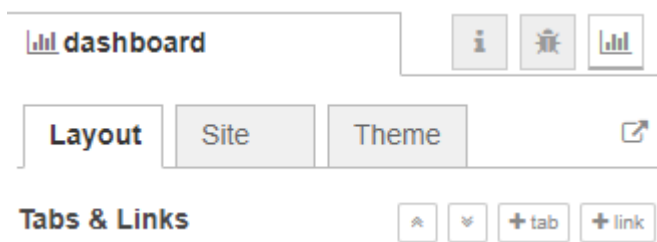
```
leur", "tab": "a740d7bd.dc5c68", "order": 1, "disp": true, "width": "6", "collapse": false}, {"id": "a740d7bd.dc5c68", "type": "ui_tab", "z": "", "name": "Test", "icon": "dashboard", "order": 1, "disabled": false, "hidden": false}]
```

• Principe


Les nœuds de l'onglet **dashboard** fournissent des widgets qui apparaissent dans l'interface utilisateur de l'application.

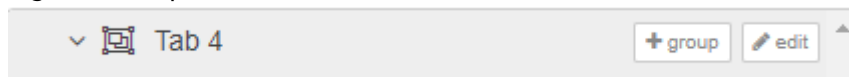
L'interface utilisateur est organisée en onglets (**Tabs**) et en groupe (**group**). Les **onglets** sont les différentes **pages** de l'interface, comme plusieurs onglets dans un navigateur. Dans chaque onglet, vous avez des **groupes** qui divisent les onglets en **sections** afin d'organiser les widgets.

Chaque widget doit être associé à un groupe qui déterminera où le widget doit apparaître sur l'interface utilisateur.

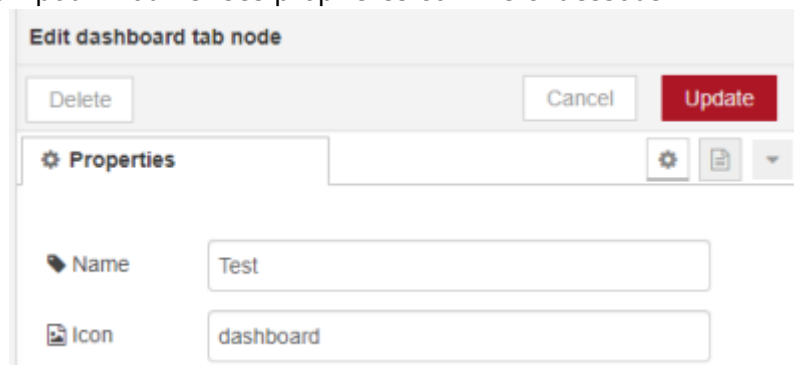


• Création de la page "Test"

1. **Sélectionnez** l'icône **dashboard**  située dans le coin supérieur droit de la fenêtre Node-RED. L'onglet dashboard (tableau de bord) apparaît comme ci-contre.
2. **Ajoutez** un onglet en cliquant sur **+tab**.



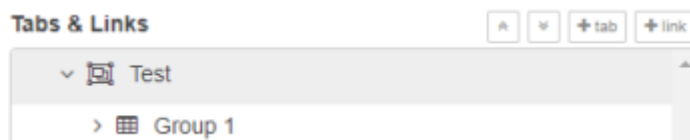
3. **Cliquez** sur edit pour modifier ses propriétés comme ci-dessous.



le nom de l'icône doit être choisie dans la liste disponible [ici](#).

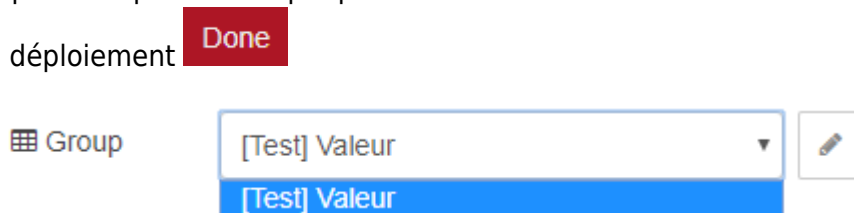


Après avoir créé un onglet, vous pouvez créer plusieurs groupes sous cet onglet. Vous devez créer au moins un groupe pour ajouter vos widgets.



- **Création de la section "Valeur" dans la page "Test"**

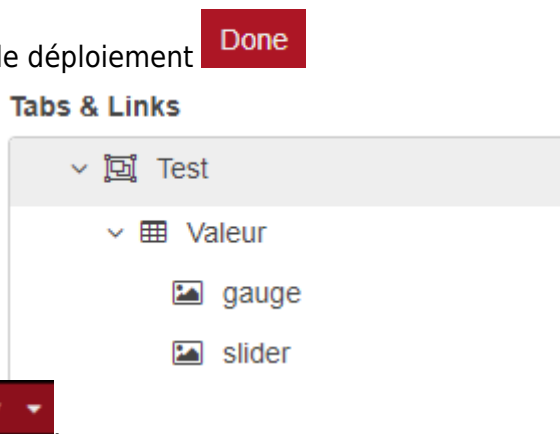
1. Cliquez sur **+group** dans l'onglet *Test*.
2. **Editez** le nom du groupe pour remplacer Group 1 par valeur.
3. Cliquez sur le bouton de déploiement



- **Association des noeuds au groupe "Valeur"**

Les noeuds *slider* et *gauge* doivent être placés dans le groupe Valeur pour que les widgets apparaissent sur la page.

1. Pour cela, **double-cliquez** sur chacun des noeuds et réglez le paramètre *Group* sur [Test]Valeur.
2. Cliquez sur le bouton de déploiement



3. Cliquez sur

- **Affichage du résultat**

Entrez `<@IPRaspi>:1880/ui` dans un navigateur. La page présentée dans l'objectif doit apparaître. Déplacez le curseur, la jauge doit réagir.

- **Code de l'exemple**

*.js

```
[{"id":"21b65a95.c084f6","type":"ui_gauge","z":"66098dac.f28304","name":"","group":"84281dde.b3e05","order":1,"width":0,"height":0,"gtype":"gauge","title":"gauge","label":"units","format":"{{value}}","min":0,"max":10,"colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":420,"y":220,"wires":[]},{ "id":"84281dde.b3e05","type":"ui_group","name":"valeur","tab":"f28a60dc.9887","order":1,"disp":true,"width":6,"collapse":false},{ "id":"f28a60dc.9887","type":"ui_tab","name":"Test","icon":"dashboard","order":1,"disabled":false,"hidden":false}]
```

8. Sécuriser Node-Red

- **Source** : [Securing Node-RED](#)



9. Projet Domotique HG (MàJ le 3/11/21)



- Description du projet [ici](#)
- Code des cartes ArduinoESP8266, ESP32 et MKR1010 [ici](#)
- Code Node-Red [ici](#)

1)

Les nœuds sont déclenchés soit en recevant un message du nœud précédent dans un flux, soit en attendant un événement externe, tel qu'une demande HTTP entrante, une temporisation ou une modification matérielle GPIO. Ils traitent ce message, ou cet événement, puis peuvent envoyer un message aux nœuds suivants du flux.

2)

Un flux est représenté sous la forme d'un onglet dans l'espace de travail de l'éditeur et constitue le moyen principal d'organiser les nœuds.

3)

Les messages sont ce qui passe entre les nœuds d'un flux. Ce sont des objets JavaScript simples qui peuvent avoir n'importe quel ensemble de propriétés. Ils sont souvent désignés sous le nom msg de l'éditeur.

4)

Les messages possèdent un payload, c'est à dire, une propriété contenant les informations les plus utiles.

5)

Les fils connectent les nœuds et représentent la manière dont les messages transitent.

6)

L'espace de travail est la zone principale où les flux sont développés en faisant glisser les nœuds de la palette et en les reliant ensemble.

7)

zone de l'éditeur node-RED destinée à la création de l'interface utilisateur

8)

onglet de dashboard, permet d'ajouter et de réorganiser les pages de l'interface, réorganiser les groupes et les widgets, puis ajouter et modifier leurs propriétés.

9)

section dans une page

10)

page de l'interface utilisateur

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=raspberrypi:linux:nodered&rev=1656772005>



Last update: **2022/07/02 16:26**