



# Découverte

## 351 - Programmation Node-Red

Rédigé par

**David ROUMANET**  
Professeur BTS SIO

Changement

Date	Révision

## Sommaire

A Introduction.....	1
A.1 No Code / Low Code.....	1
A.2 Prérequis.....	1
B Node-Red.....	2
B.1 Présentation.....	2
B.2 Fonctionnement.....	2
B.3 Installation et exécution.....	3
C Exemples d'applications.....	4
C.1 Application N°1 : Hello Word.....	4
C.2 Application N°2 : passage de valeur.....	5
C.3 Application N°3 : interrogation API RESTful.....	6
D Annexes.....	8
D.1 Sources.....	8
D.2 Autres.....	8
E Ce qu'il faut retenir.....	10

---

## A Introduction

---

Régulièrement, de nouvelles rumeurs apparaissent sur différentes méthodes de programmation. Je ne citerais ici que les principales :

- L'IA remplacera les codeurs
- La nouvelle manière de coder, est sans code
- Go<sup>1</sup> remplacera la plupart des langages
- Web Assembly est l'avenir de la programmation dans les navigateurs
- ...

La plupart de ces déclarations sont liées à un engouement pour de nouvelles technologies. Mais dans cette découverte, nous allons nous intéresser au codage sans code (No Code ou Low Code).

### A.1 No Code / Low Code

Depuis plusieurs années, on rencontre cette nouvelle méthode de programmation, dite graphique.

L'objectif de ces outils NoCode est de permettre de transformer la phase de codage en phase de gestion de processus, pour simplifier le travail du développeur, mais surtout, diminuer les coûts de développement.

Pour cela, il faut utiliser des outils graphiques permettant de relier des cellules spécialisées dans lesquelles on saisit le minimum d'informations.

Toutefois, ce genre de plateforme a également ses inconvénients : le code devient moins accessible, le développeur est plus dépendant de l'outil.

Par exemple, en Javascript, vous pouvez utiliser l'éditeur de votre choix, comme VSCode, Notepad++, PHPStorm, NetBean... ce n'est plus le cas en nocode).

Nous allons donc aborder de manière très légère ce concept, avec l'environnement Node-Red qui est Low code (nécessite encore quelques connaissances), afin d'avoir une culture et une compréhension globale du fonctionnement.

### A.2 Prérequis

Pour cette approche "découverte", Il faut maîtriser les concepts suivants :

- NodeJS et installation de framework avec NPM
- les API RESTFul
- JavaScript

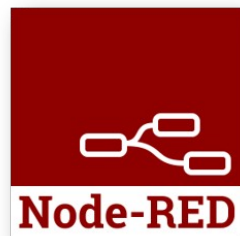
---

1 Remplacez par votre langage préféré : Rust, ADA, PHP, etc.

## B Node-Red

### B.1 Présentation

Node-Red est un langage de développement graphique créé par IBM, au début de l'année 2013.

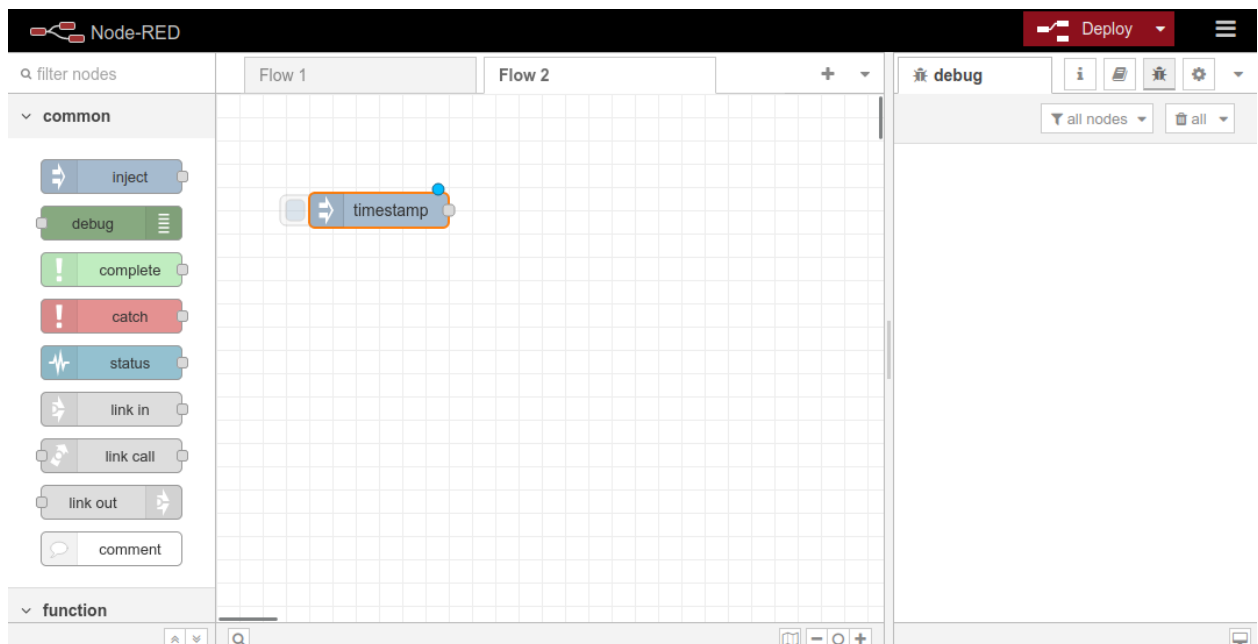


Son objectif est de relier des équipements matériels, des API et des services en ligne pour faciliter l'intégration des IoT (Internet of things).

Il fait désormais partie de la fondation JS depuis 2016 et est diffusé en open-source.

### B.2 Fonctionnement

L'éditeur graphique est un service disponible dans un navigateur Internet, mais l'ensemble s'appuie sur NodeJS pour l'exécution (côté serveur).



L'importation et l'exportation de programmes se fait au format JSON.

Note-Red propose plusieurs types de blocs :

- Les blocs **communs** sont les plus fréquemment rencontrés
- Les blocs **fonctions** sont programmables
- Les blocs **réseaux** permettent la communication avec le réseau
- Les blocs **séquences** sont des blocs qui facilitent le traitement de données (joindre, séparer, trier...)
- Les blocs d'**analyses** travaillent sur la conversion des formats de données (CSV, JSON, XML...)
- Les blocs de **stockage** servent pour la gestion des fichiers

### B.3 Installation et exécution

L'installation de Node-Red se fait avec l'instruction `npm install -g --unsafe-perm node-red`.

L'exécution de la commande `node-red` lance un serveur sur le port local 1880 :

```
node-red
16 Oct 22:41:53 - [info]
Welcome to Node-RED
=====
16 Oct 22:41:53 - [info] Node-RED version: v3.0.2
16 Oct 22:41:53 - [info] Node.js version: v16.16.0
16 Oct 22:41:53 - [info] Linux 5.19.14-1-MANJARO x64 LE
16 Oct 22:41:53 - [info] Loading palette nodes
16 Oct 22:41:53 - [info] Settings file : /home/david/.node-red/settings.js
16 Oct 22:41:53 - [info] Context store : 'default' [module=memory]
16 Oct 22:41:53 - [info] User directory : /home/david/.node-red
16 Oct 22:41:53 - [warn] Projects disabled : editorTheme.projects.enabled=false
16 Oct 22:41:53 - [info] Flows file : /home/david/.node-red/flows.json
16 Oct 22:41:53 - [info] Server now running at http://127.0.0.1:1880/
16 Oct 22:41:53 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

16 Oct 22:41:53 - [info] Starting flows
16 Oct 22:41:53 - [info] Started flows
```

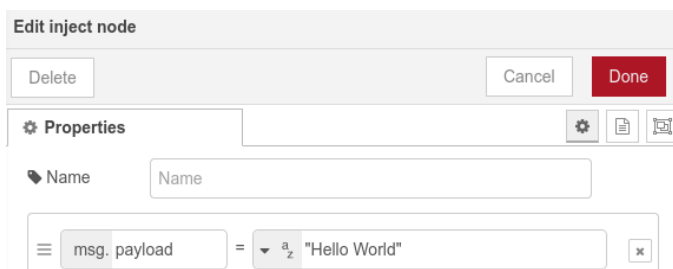
Un rappel de sécurité indique que node-red n'est pas sécurisé (section orange ci-dessus).

## C Exemples d'applications

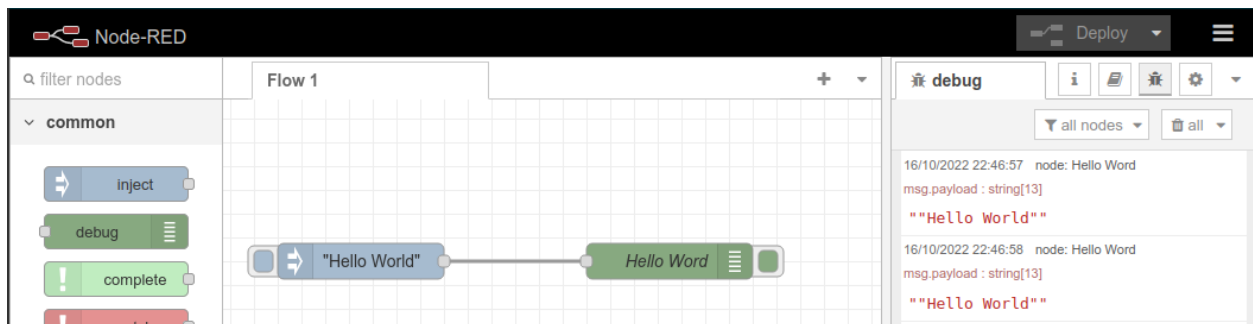
### C.1 Application N°1 : Hello Word

La première application simple est l'affichage d'un "Hello World" dans la console de débogage. Pour cela, vous devez créer 5 choses :

- Faire un glisser/déposer du symbole `common → inject` sur la feuille centrale [Flow 1]
- Un double-clic sur ce symbole pour créer la chaîne de caractères (string) "Hello World"



- Faire un glisser/déposer du symbole `common → debug` à droite du symbole inject "Hello World"
- Un double-clic pour renommer le débogueur en "Hello World"
- Créer un lien entre les deux points de ces deux symboles, comme ci-dessous.

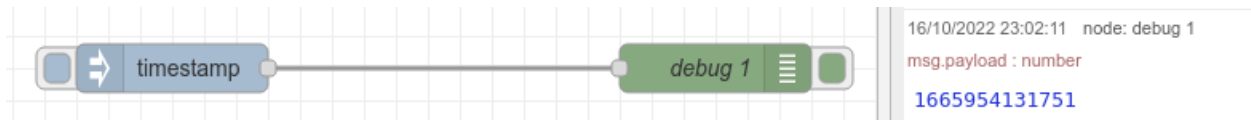


Activez votre programme en cliquant sur le bouton rouge `Deploy` en haut à droite, puis cliquez sur l'onglet [Debug] 

Enfin, Cliquez sur la zone bleue à gauche du symbole injecteur "Hello World" : la fenêtre de débogage doit afficher un message à chaque clic.

## C.2 Application N°2 : passage de valeur

Il s'agit de relier à nouveau un `common → inject` (pas de configuration, on garde TimeStamp par défaut) et un `common → debug` par défaut.



Le nombre qui s'affiche à chaque clic est un estampillage horaire (en secondes). Nous allons insérer une fonction qui va convertir le timestamp en format date avant de l'envoyer dans la console de débogage.

Faites glisser un objet `function → function` entre les deux objets précédents (avec un peu de chance, le lien change en pointillé, lâchez l'objet). Si cela ne fonctionne pas, débranchez le lien et connectez-le à la nouvelle fonction, puis créez un nouveau lien vers l'autre objet.



Programmez ensuite l'objet fonction en double-cliquant dessus, comme suit :

name : conversion

Onglet [onMessage]

```
let date = new Date(msg.payload);
msg.payload = date.toString();
return msg;
```

Validez puis relancez le déploiement de la solution (bouton `Deploy`). Désormais, l'affichage dans la console doit changer à chaque nouveau clic sur l'objet 'timestamp'.



Par défaut, les objets génèrent des messages dans un objet au format JSON, nommé msg.

Msg.payload contient donc le message injecté par l'objet `common → inject`.

---

Essayez d'utiliser la fonction `.split(" ")` pour n'afficher que l'heure au format HH:MM:SS

---

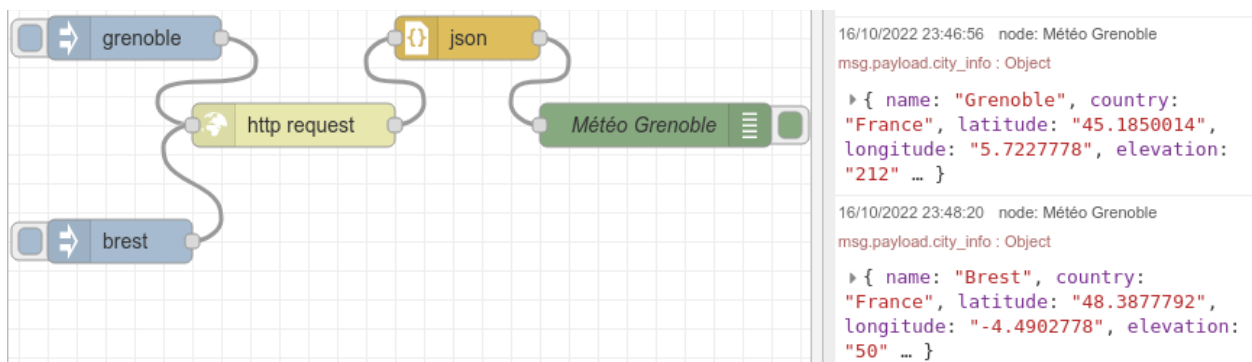
### C.3 Application N°3 : interrogation API RESTful

Node-Red est évidemment bien plus puissant et nous allons interroger la météo du site <https://www.prevision-meteo.ch/services/json/>

Pour fonctionner, notre application va transmettre une chaîne de caractère (la ville dont nous voulons avoir la météo) à un nouvel objet : `network → http request`. La réponse étant au format JSON, nous utiliserons également un objet `parser → json`.

- Dans l'objet `common → inject`, modifiez la valeur du `msg.payload` par une chaîne "grenoble"
- Dans l'objet `network → http request` ajouter `{{payload}}` après l'url du service API (URL = <https://www.prevision-meteo.ch/services/json/{{payload}}>)
- Dans l'objet `parser → json`, choisissez l'action "Convert between JSON String and Object"
- Dans l'objet `common → debug`, modifier l'affichage pour une partie de l'objet : `msg.payload.city_info`

Reliez les éléments entre eux comme ci-dessous et déployez pour valider le fonctionnement.



Dans l'exemple ci-dessus, vous pouvez constater que nous pouvons utiliser plusieurs injecteurs sur un même objet.

---

Sauriez-vous afficher la ville et sa température courante (`current_condition`) avec deux debug ?

---





## **D Annexes**

---

### **D.1 Sources**

### **D.2 Autres**

## À EFFACER LORSQUE LE DOCUMENT EST TERMINÉ

Un encart de base

---

Attention, il est important de vérifier blabla que blabla. Cela peut arriver lorsque blabla ou bien que truc active le machin et dépasse la limite du bidule. Il est donc important de ne pas laisser les choses dériver sans un contrôle des valeurs sinon blabla.

Il reste cependant la possibilité d'écrire en dessous du nombre de lignes.

---

Un encart d'attention + puce attention



Attention, il est important de vérifier blabla que blabla. Cela peut arriver lorsque blabla ou bien que truc active le machin et dépasse la limite du bidule. Il est donc important de ne pas laisser les choses dériver sans un contrôle des valeurs sinon blabla. Il reste cependant la possibilité d'écrire en dessous du nombre de lignes.

---

Un encart stop + puce stop



Attention, il est important de vérifier blabla que blabla. Cela peut arriver lorsque blabla ou bien que truc active le machin et dépasse la limite du bidule. Il est donc important de ne pas laisser les choses dériver sans un contrôle des valeurs sinon blabla. Il reste cependant la possibilité d'écrire en dessous du nombre de lignes.

---

Un encart de réflexion



Attention, il est important de vérifier blabla que blabla. Cela peut arriver lorsque blabla ou bien que truc active le machin et dépasse la limite du bidule. Il est donc important de ne pas laisser les choses dériver sans un contrôle des valeurs sinon blabla. Il reste cependant la possibilité d'écrire en dessous du nombre de lignes.

---

Un encart de note



Attention, il est important de vérifier blabla que blabla. Cela peut arriver lorsque blabla ou bien que truc active le machin et dépasse la limite du bidule. Il est donc important de ne pas laisser les choses dériver sans un contrôle des valeurs sinon blabla.

Il reste cependant la possibilité d'écrire en dessous du nombre de lignes.

---

---

## E Ce qu'il faut retenir...

---

Insertion > Autres sauts > Saut Manuel...

Choisir le style de page « Résumé »

Attention : résumé s'enchaîne avec Style de page droite ► Mettre une page « Résumé » à la place d'une page Gauche.

Les liens sont sous et surlignés