Doc Node Red en FR

Doc EN Node-Red

Guide de programmation Node RED

Programmation de l'IoT

Node-RED : Conférence 1 – Une brève introduction à Node-RED

Node-RED est un outil puissant pour créer des applications Internet des objets (IoT) en mettant l'accent sur la simplification du “câblage” des blocs de code pour effectuer des tâches. Il utilise une approche de programmation visuelle qui permetaux développeurs de connecter des blocs de code prédéfinis, appelés « nœuds », pour effectuer une tâche. Les nœuds connectés, généralement une combinaison de nœuds d'entrée, de nœuds de traitement et de nœuds de sortie, lorsqu'ils sont câblés ensemble, constituent un « flux ».

Développé à l'origine en tant que projet open source chez IBM fin 2013, pour répondre à leur besoin de connecter rapidement du matériel et des appareils à des services Web et à d'autres logiciels - comme une sorte de colle pour l'IoT - il a rapidement évolué pour devenir une programmation IoT à usage général. outil. Il est important de noter que Node-RED a rapidement développé une base d'utilisateurs importante et croissante et une communauté de développeurs active qui contribuent à de nouveaux nœuds permettant aux programmeurs de réutiliser le code Node-RED pour une grande variété de tâches.

Bien que Node-RED ait été conçu à l'origine pour fonctionner avec l'Internet des objets, c'est-à-dire des appareils qui interagissent et contrôlent le monde réel, au fur et à mesure de son évolution, il est devenu utile pour une gamme d'applications.

Cette conférence vous présentera quelques exemples de Node-RED, à la fois pour vous donner une idée de ses capacités et pour introduire ses concepts de base. Nous discuterons également de l'histoire de Node-RED et de ses origines pour vous donner une idée de ce à quoi Node-RED est bon et des types de tâches qu'il vaut mieux laisser à un langage et à des environnements de programmation plus traditionnels.

À la fin de cette conférence, vous devriez avoir une compréhension de haut niveau de Node-RED, de son modèle de programmation et de quelques nœuds simples prédéfinis, ainsi qu'un aperçu des origines et de la meilleure utilisation de Node-RED. Apprendre à connaître Node-RED

Commençons par regarder quelques exemples simples des types de choses que vous pouvez faire avec Node-RED. Notre objectif dans cette conférence n'est pas de vous faire programmer avec Node-RED, mais de vous donner une idée de ses capacités à travers quelques exemples de haut niveau. Si vous voulez vraiment plonger directement, vous pouvez passer directement à la conférence 2 où vous serez guidé à travers les étapes pour créer et exécuter votre premier flux.

Ce premier exemple de flux recevra un message d'un flux Twitter contenant un hashtag et allumera et éteindra une LED lorsqu'un nouveau tweet est arrivé. Pour cet exemple, supposons que NODE-RED fonctionne sur un Raspberry PI - une situation courante pour de nombreux utilisateurs de Node-RED.

Exemple 1.1 Utilisation de Twitter pour contrôler un Raspberry PI

Le flux Node-RED pour vérifier Twitter et pour voir si un hashtag #led est détecté, allume une LED connectée à un Pi est illustré à la Fig. 1.1. Comme vous pouvez le voir, c'est assez simple; il se compose de 3 nœuds câblés ensemble - un nœud twitter à gauche, un nœud de déclenchement au milieu et un nœud Raspberry Pi (gpionode) à droite. (gpio signifie General Purpose Input/Output et est le nom donné au nœud générique Raspberry PI qui gère les E/S). Les nœuds, représentés par des cases arrondies, ont généralement une icône pour représenter leur type, par exemple l'icône d'oiseau Twitter dans le premier nœud, et un nom de texte qui peut être défini par le programmeur.

Chacun de ces trois nœuds est intégré à l'outil de programmation visuelle Node-RED et peut être déplacé d'une palette de nœuds vers l'espace de travail principal. Ils sont ensuite “câblés” ensemble en joignant leurs onglets de sortie à l'onglet d'entrée du nœud suivant dans le flux (les petites gouttes grises à droite ou à gauche des boîtes de nœud.) La prochaine session examinera l'interface utilisateur en détail, mais pour l'instant concentrons-nous sur les capacités de haut niveau.

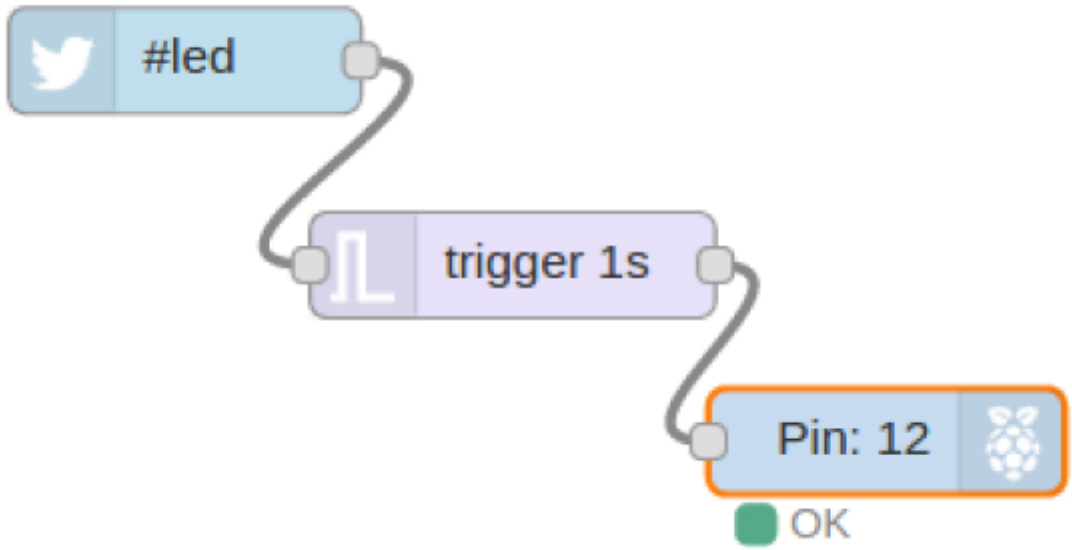


Figure 1.1 Un flux simple à 3 nœuds qui utilise une balise Twitter pour déclencher une LED sur un Raspberry PI.

Le nœud Twitter est un nœud intégré dans Node-RED et cache toute la complexité de l'utilisation de l'API Twitter. Il peut être configuré avec les informations d'identification du compte d'un utilisateur et une variété de chaînes de recherche, dans notre cas, il recherche simplement le hashtag '#led'.

Lorsque le nœud Twitter voit la balise soit dans un tweet à l'utilisateur, soit dans le flux de tweet public, il crée un nouveau message avec les détails du tweet, qui est transmis au nœud suivant dans

le flux. Dans notre exemple, puisque la sortie du nœud Twitter est connectée à l'entrée du nœud déclencheur, le nœud déclencheur reçoit un message avec les détails du tweet.

Le nœud déclencheur est un autre nœud intégré dans Node-RED et son comportement par défaut est d'attendre tout message sur son entrée. Lorsqu'il reçoit un message, il se « déclenche » et envoie un message sur sa sortie avec la valeur « 1 » comme corps du message. Il attend ensuite 1 seconde et envoie un second message avec la valeur « 0 » dans le corps du message.

Étant donné que le nœud de déclenchement est connecté au gpionode, qui contrôle les broches d'entrée/sortie ou IO sur le Raspberry PI, il reçoit ces deux messages, à 1 seconde d'intervalle, sur son entrée.

Encore une fois, le gpionode est un nœud intégré dans Node-RED qui masque les détails de la configuration des broches IO sur le PI. Lorsqu'il reçoit un message avec une valeur de “1” dans le corps, il prend la broche IO haute, et lorsqu'il reçoit un message avec un corps de “0”, il prend la broche basse. Dans cet exemple, le gpionode est configuré pour contrôler la broche numéro 12, et le nœud l'affiche donc dans l'icône du nœud.

Si vous avez votre PI câblé avec une LED connectée à la broche 12, le gpionode passant au niveau haut pendant 1 seconde puis au niveau bas fera clignoter la LED pendant 1 seconde.

Si vous pensez à ce qui se passe ici, c'est en fait un exemple incroyablement puissant de Node-RED et de l'Internet des objets (IoT). Il connecte le “Twitterverse” à un PI et à une LED, mais ne nécessite aucune programmation - toutes les tâches sont effectuées par des nœuds pré-construits dans Node-RED qui ont juste besoin d'être configurés. Cela démontre vraiment la puissance de Node-RED lorsqu'il est exécuté sur un appareil simple tel qu'un Raspberry PI.

Alors juste pour récapituler. Node-RED est un outil de programmation basé sur les flux avec un éditeur visuel qui vous permet de relier des nœuds pour créer des flux. Dans ce cas, 3 nœuds créent notre premier flux. Les événements du monde réel, par exemple un tweet avec le hashtag #led, sont convertis en messages, qui « circulent » le long des fils entre les nœuds. Les nœuds traitent les messages sur leur entrée et, par conséquent, envoient des messages sur leurs sorties au nœud suivant dans le flux. Dans cet exemple simple, nous recherchons un tweet avec le hashtag #led et l'utilisons pour déclencher un message vers un nœud Raspberry PI qui fait monter la broche IO 12 pendant 1 seconde, ce qui à son tour fait clignoter une LED pendant 1 seconde .

Cet exemple - et le reste de cette conférence - est principalement destiné à vous montrer le genre de choses que nous pouvons faire avec Node-RED. Si vous voulez jouer avec l'exemple, vous devriez regarder la leçon 2 pour voir comment configurer un compte avec FRED (le service Node-RED basé sur le cloud que nous utilisons) et comment commencer. Vous pouvez trouver la description node-red de ce flux sur :

https://raw.githubusercontent.com/SenseTecnic/nrguideflows/master/lesson1/1-1_twitter.json

Exemple 1.2 Utilisation de Node-RED pour vous alerter lorsque vous prenez du retard sur votre programme d'exercices

Notre deuxième exemple montre une utilisation différente de Node-RED. Plutôt que de contrôler un appareil tel qu'un Raspberry PI, ce flux combine les informations météorologiques et votre utilisation de Fitbit pour vous envoyer une alerte par e-mail lorsque la météo est bonne et que vous êtes en retard sur votre programme d'exercice. Le flux est un peu plus compliqué, utilisant 6 nœuds (voir Fig

1.2), mais encore une fois, en utilisant des nœuds intégrés, vous pouvez accomplir des tâches complexes avec peu ou pas de programmation.

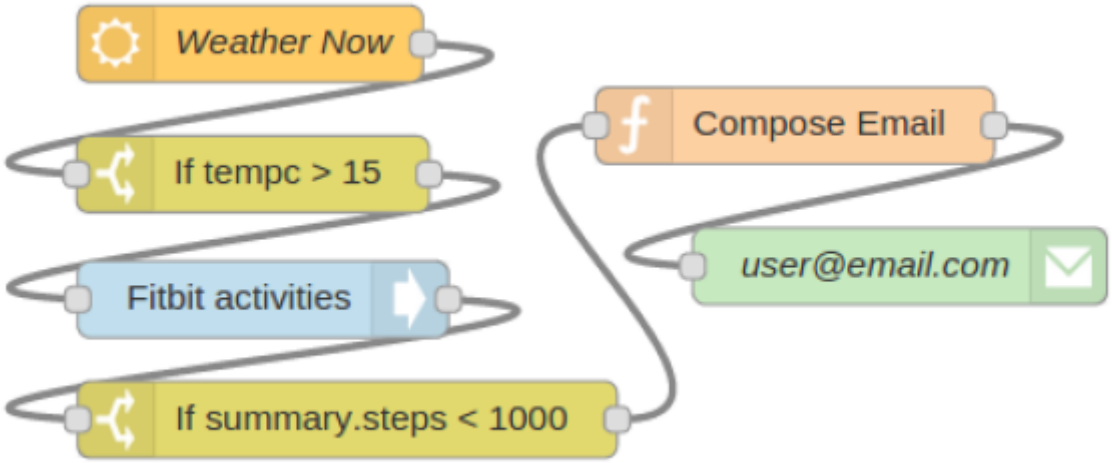


Figure 1.2 Un flux simple pour vous alerter lorsque vous ne faites pas d'exercice et qu'il fait beau

Comme auparavant, les nœuds sont glissés depuis la palette de nœuds dans l'interface utilisateur Node-RED et câblés ensemble sur le canevas de flux. En commençant par le nœud le plus à gauche, commençons par un nœud openweather que vous pouvez configurer pour vérifier la météo de n'importe quelle grande ville à des intervalles définis. Lorsqu'il vérifie, il regroupe les résultats dans un message qui est transmis au nœud suivant, dans ce cas, à un nœud de commutation. Le nœud de commutation effectue un test simple, vérifiant le message d'entrée pour une valeur et envoyant un message avec une valeur de « 1 » si le test est vrai. Dans notre cas, il a été configuré pour tester la valeur de température (tempc) pour voir si elle est de 15 degrés Celsius ou plus.

Lorsque le nœud de commutation teste vrai ou « s'allume », il envoie un message au nœud fitbit qui est le nœud suivant dans le flux. Le nœud fitbit est un autre nœud intégré puissant que vous pouvez configurer pour obtenir des statistiques sur votre appareil fitbit à partir du portail fitbit.com. Le nœud Fitbit regroupe les statistiques de votre appareil dans un message et le transmet au nœud suivant du flux.

Le nœud suivant est un autre nœud de commutation. Celui-ci a été configuré pour vérifier si le nombre de pas que vous avez parcourus aujourd'hui est inférieur à 1000. Il le fait en testant pour voir si le champ de données fitbit, summary.steps, est inférieur à 1000. N'oubliez pas qu'il vient de recevoir ces statistiques dans un message du nœud fitbit, qui à son tour les a obtenues en appelant l'API fitbit sur fitbit.com.

Si vous avez parcouru moins de 1 000 pas aujourd'hui, le nœud de commutation génère un message qui est transmis via le flux à un nœud de fonction. Le nœud de fonction est un nœud générique polyvalent qui vous permet d'écrire votre propre logique de programmation en Javascript. Ce nœud sera discuté plus en détail dans la prochaine leçon. Pour l'instant, vous pouvez supposer que ce nœud crée simplement un nouveau message avec une ligne d'objet d'e-mail et un corps de texte pouvant être envoyé par e-mail.

Une fois ce message créé dans le nœud de fonction, il est envoyé via le flux au nœud final qui est un nœud de messagerie. Ce nœud prendra le corps de tout message entrant et l'enverra au compte de messagerie configuré lors de l'initialisation du nœud. Dans ce cas, vous le configureriez pour vous envoyer un message.

Encore une fois, un flux très simple qui utilise la puissance des nœuds intégrés de Node-RED pour vérifier votre météo locale. Et s'il fait raisonnablement chaud et que vous n'avez pas beaucoup marché aujourd'hui, il vous envoie un rappel par e-mail.

Bien que l'exemple semble assez simple, il suffit de vérifier la météo et votre activité fitbit, vous pouvez utiliser des flux similaires pour vérifier l'état des appareils domestiques, des ordinateurs de votre batterie de serveurs, des appareils dans une usine, etc., et vous pouvez ensuite utiliser nœuds de sortie pour envoyer du courrier, des tweets, effectuer des appels d'API vers des logiciels de back-office, contrôler des appareils du monde réel - en fait, les choix sont infinis. Node-RED est un outil extrêmement puissant pour relier les entrées et les sorties et offre de nombreux nœuds, intégrés ou développés par la communauté Node-RED, pour effectuer une gamme incroyable de tâches.

Vous pouvez trouver la description node-red de ce flux sur :

https://raw.githubusercontent.com/SenseTecnic/nrguideflows/master/lesson1/1-2_weatheralert.json

Exemple 1.3 Création d'un service Web simple à l'aide des nœuds HTTP intégrés de Node-RED

Encore une fois, c'est juste pour vous donner une idée de la puissance et de la flexibilité de Node-RED sans entrer dans tous les détails du fonctionnement de ce flux. Vous en saurez plus à ce sujet dans des conférences ultérieures. À un niveau élevé, le nœud d'entrée http a été configuré pour écouter les requêtes HTTP sur une URL, qui est une combinaison du service d'hébergement utilisé dans ces cours, plus votre nom de connexion {username} et la chaîne /fitbit. Ceci sera expliqué plus en détail dans la deuxième leçon.

Lorsqu'une requête HTTP arrive, le nœud d'entrée http crée un message pour déclencher le nœud fitbit, qui est le nœud suivant dans le flux. Le nœud fitbit obtient les statistiques actuelles de l'utilisateur, puis transmet ces statistiques, sous forme de message, à un nœud de modèle. Le nœud de modèle html est un autre nœud pré-construit dans Node-RED, qui, comme le nœud de fonction, vous permet de créer du code arbitraire. Cependant, plutôt que d'utiliser JavaScript comme le nœud de fonction, le nœud de modèle fonctionne avec du texte tel que HTML.

Le code HTML du nœud de modèle est présenté dans le Listing 1.1, vous pouvez donc voir ce qui se passe. Cependant, il n'est pas nécessaire de l'examiner en détail si vous ne le souhaitez pas. Fondamentalement, le nœud de modèle prend les données de consommation de calories de fitbit et, à l'aide d'une bibliothèque graphique tierce, crée un fichier HTML simple qui indiquera au navigateur de réception d'utiliser la bibliothèque graphique tierce pour afficher un graphique en anneau de la consommation de calories.

Listing 1.1 Un modèle HTML simple pour afficher un graphique en anneau de l'utilisation des calories

modelhtml.html

```
    <!doctype html>
    <head>
        <title>A Node RED Example</title>
        <link rel="stylesheet"
href="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.css">
        <script
src="//cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-
min.js"></script>
        <script
src="//ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.min.js"></scri
pt>
        <script
src="//cdnjs.cloudflare.com/ajax/libs/morris.js/0.5.1/morris.min.js"></
script>
    </head>
    <html>
        <div id="chart-example" style="height: 250px;"></div>
        <script>
            Morris.Donut({
               element: 'chart-example',
               data: [
                 {label: "Activity Calories", value:
{{payload.summary.activityCalories}} },
                 {label: "Calories BMR", value:
{{payload.summary.caloriesBMR}} },
                 {label: "Calories Out", value:
{{payload.summary.caloriesOut}} }
               ]
            });
        </script>
    </html>
```

Une fois que le nœud de modèle a généré le fichier HTML, il le transmet sous forme de message au nœud final, qui est un nœud de réponse http. Ce nœud regroupe le code HTML sous la forme d'une réponse HTTP qu'il renvoie au navigateur.

L'utilisateur verra alors un simple graphique en anneau de sa consommation de calories pour la journée (voir Fig 1.4) - le tout construit et servi par un simple flux Node-RED !

Figure 1.4 Un graphique en anneau, servi par Node-RED montrant le nombre de calories de fitbit

Vous pouvez trouver la description node-red de ce flux sur :

https://raw.githubusercontent.com/SenseTecnic/nrguideflows/master/lesson1/1-3_webservice.json

L'histoire de Node-RED

Comme les exemples de la section précédente l'ont montré, Node-RED est un outil puissant pour créer des applications et des services IoT. Sa genèse a été déclenchée précisément par ce besoin de prototyper rapidement des applications IoT et Node-RED a été créé en tant que projet open source par le groupe IBM Emerging Technologies, notamment par deux chercheurs, Nick O'Leary et Dave Conway-Jones. Ils ont d'abord créé Node-RED comme un outil pour eux-mêmes alors qu'ils travaillaient sur des projets IoT et «recherchaient un moyen de simplifier le processus de connexion des systèmes et des capteurs lors de la création de technologies de preuve de concept pour les clients».

Une première version de Node-RED a été publiée en tant que projet open source à la fin de 2013 et a constitué un petit groupe d'utilisateurs et de développeurs actifs en 2014. Au moment de la rédaction de cet article, Node-RED est encore une technologie émergente, mais a déjà vu adoption significative par les fabricants, les expérimentateurs et un certain nombre de grandes et petites entreprises qui expérimentent son utilisation pour leurs besoins.

Aujourd'hui, il existe une communauté dynamique d'utilisateurs et de développeurs, avec un groupe central travaillant sur le code Node-RED lui-même et la plupart des développeurs contribuant aux

nœuds ou aux flux de la bibliothèque de flux. Vous pouvez consulter la liste de diffusion et les référentiels github en utilisant les liens fournis dans l'annexe A ou simplement les rechercher sur Google.

Étant donné que Node-RED est encore une technologie en évolution rapide, sachez que les choses peuvent changer rapidement. Ces cours ont été rédigés par rapport à la version 0.11.0 de Node-RED et les exemples seront mis à jour au fur et à mesure de l'évolution de Node-RED. Cependant, il est toujours sage, comme pour toute nouvelle technologie, de vérifier la compatibilité si les choses ne fonctionnent pas comme prévu.

Node-RED et l'IoT

Lorsque les gens d'IBM ont créé Node-RED, ils se concentraient principalement sur l'Internet des objets, c'est-à-dire la connexion des appareils au traitement et le traitement aux appareils. En tant qu'outil de développement rapide d'applications pour l'IoT, Node-RED est à la fois puissant et flexible. Sa puissance provient de la combinaison de deux facteurs :

- Node-RED est un exemple de modèle de programmation basé sur les flux - des messages représentant des événements circulent entre les nœuds, déclenchant un traitement qui aboutit à une sortie. Le modèle de programmation basé sur les flux correspond bien aux applications IoT typiques qui se caractérisent par des événements du monde réel qui déclenchent une sorte de traitement qui à son tour entraîne des actions du monde réel. Node-RED regroupe ces événements sous forme de messages qui offrent un modèle simple et uniforme pour les événements lorsqu'ils circulent entre les nœuds qui composent les flux.
- L'ensemble des nœuds intégrés est la deuxième partie de l'histoire de Node-RED. En construisant un ensemble de puissants nœuds d'entrée et de sortie, dont chacun cache une grande partie de la complexité de l'interaction avec le monde réel, Node-RED offre aux développeurs de puissants blocs de construction pour leur permettre de mettre rapidement en place des flux qui accomplissent beaucoup, sans avoir à s'inquiéter des détails de la programmation.

Ces deux facteurs font de Node-RED un outil puissant pour le développement d'applications IoT. Lorsqu'il est combiné avec la flexibilité de créer et d'utiliser des nœuds tels que le nœud de fonction, qui permet au développeur d'écrire rapidement du JavaScript arbitraire, ainsi que la communauté Node-RED, qui crée et partage constamment de nouveaux nœuds, Node-RED est susceptible d'être un des principaux outils de la boîte à outils des développeurs IoT.

Cependant, Node-RED n'est pas toujours le meilleur outil pour le développement d'applications IoT. Bien que puissant et flexible, ce n'est pas toujours la bonne solution. Certaines situations où Node-RED peut ne pas être le premier choix incluent :

- Applications IoT multifonctions complexes. Node-RED excelle dans le développement rapide d'applications et sert de ciment pour connecter des événements à des actions, ou des capteurs à des actionneurs, si vous le souhaitez. Cependant, lorsqu'une application dépasse une certaine taille, elle devient complexe à programmer visuellement et à gérer via Node-RED. Il existe certaines fonctionnalités pour aider à cela, par exemple les sous-flux (voir leçon 5), mais finalement l'interface utilisateur devient un goulot d'étranglement.
- La programmation basée sur les flux est un modèle de programmation qui n'est pas toujours le plus adapté au développement d'applications. De la même manière que certains langages de programmation excellent dans certaines tâches mais pas dans d'autres, la programmation

basée sur les flux a ses faiblesses. Un bon exemple sont les boucles : Node-RED est lourd lors de la gestion des boucles.
- Cas d'utilisation spécifiques. La programmation basée sur les flux est un modèle à usage général et n'est pas ciblée ou optimisée pour des besoins spécifiques, par exemple l'analyse de données ou le développement d'interface utilisateur. Pour le moment, Node-RED n'a pas de support spécifique pour ce type d'applications et aucun moyen simple d'ajouter un tel support. Évidemment, comme la technologie sous-jacente de Node-RED est JavaScript, vous pouvez tirer parti de ses capacités pour ces besoins. Si vous souhaitez utiliser Node-RED pour prototyper quelque chose, une approche consiste à implémenter des parties ou la totalité de votre application dans un autre langage plus adapté à votre tâche et à utiliser Node-RED pour le contrôle global.

Cela dit, comme cela a été montré dans cette conférence et sera démontré dans les conférences suivantes, Node-RED est un outil puissant pour un grand nombre d'applications IoT. Au fur et à mesure de son évolution, il deviendra à la fois plus sophistiqué et plus utilisable dans un plus large éventail de circonstances.

Peut-être tout aussi important, en explorant Node-RED dans les conférences suivantes, vous vous rendrez compte que bien que les racines de Node-RED soient dans l'IoT, c'est un outil qui peut être utilisé pour créer une grande variété d'applications, pas seulement des applications IoT. En fait, dans ces conférences, vous verrez des exemples d'utilisation de Node-RED pour les applications Web, les applications de médias sociaux, l'intégration de back-office, la gestion des tâches informatiques, c'est-à-dire dans toute la gamme des tâches informatiques modernes.

Notre espoir est qu'à la fin de ce cours magistral, vous verrez Node-RED, comme nous, comme un outil flexible et puissant qui peut être utilisé dans de nombreuses situations, à la fois pour le prototypage mais aussi pour le développement au niveau de la production.

Node-RED : Cours 2 – Construire vos premiers flux

Cette conférence vous présentera l'outil visuel Node-RED et vous permettra de commencer à créer votre premier flux. Vous apprendrez à créer des flux simples, à utiliser des nœuds de débogage pour suivre les messages dans les flux et à utiliser lenœud de fonction pour écrire un code JavaScript simple qui adapte le nœud à vos besoins spécifiques.

Pour commencer, vous utiliserez un service cloud gratuit qui vous fournit une configuration Node-RED préconfigurée et une large gamme de nœuds intégrés. Des conférences ultérieures expliqueront comment configurer Node-RED par vous-même sur des appareils tels qu'un Raspberry PI.

À la fin de cette conférence, vous serez en mesure de créer vos propres flux simples, vous connaîtrez l'interface utilisateur de Node-RED et vous serez familiarisé avec certains des nœuds intégrés de base proposés par Node-RED. . FRED – Front-end pour Node-RED

FRED est un service cloud que nous avons créé qui vous permet de vous inscrire et d'utiliser Node-RED sans avoir à télécharger et à installer sur un appareil tel qu'un PI. Il facilite le démarrage avec Node-RED en gérant une instance de Node-RED pour vous, en gardant l'instance à jour avec les dernières versions de code et en appliquant des corrections de bogues ou des correctifs. Un ensemble

de nœuds supplémentaires fournis par la communauté est également inclus, qui s'ajoutent à l'ensemble de base standard que vous obtenez lorsque vous installez vous-même Node-RED. Cependant, FRED n'est qu'un wrapper pour Node-RED, donc tout ce que vous écrivez dans FRED fonctionnera sur une instance Node-RED fonctionnant sur un Pi ou un Beagleboard.

Cette conférence vous présentera l'outil visuel Node-RED et vous permettra de commencer à créer votre premier flux. Vous apprendrez à créer des flux simples, à utiliser des nœuds de débogage pour suivre les messages dans les flux et à utiliser lenœud de fonction pour écrire un code JavaScript simple qui adapte le nœud à vos besoins spécifiques.

Pour commencer, vous utiliserez un service cloud gratuit qui vous fournit une configuration Node-RED préconfigurée et une large gamme de nœuds intégrés. Des conférences ultérieures expliqueront comment configurer Node-RED par vous-même sur des appareils tels qu'un Raspberry PI.

À la fin de cette conférence, vous serez en mesure de créer vos propres flux simples, vous connaîtrez l'interface utilisateur de Node-RED et vous serez familiarisé avec certains des nœuds intégrés de base proposés par Node-RED. . FRED – Front-end pour Node-RED

FRED est un service cloud que nous avons créé qui vous permet de vous inscrire et d'utiliser Node-RED sans avoir à télécharger et à installer sur un appareil tel qu'un PI. Il facilite le démarrage avec Node-RED en gérant une instance de Node-RED pour vous, en gardant l'instance à jour avec les dernières versions de code et en appliquant des corrections de bogues ou des correctifs. Un ensemble de nœuds supplémentaires fournis par la communauté est également inclus, qui s'ajoutent à l'ensemble de base standard que vous obtenez lorsque vous installez vous-même Node-RED. Cependant, FRED n'est qu'un wrapper pour Node-RED, donc tout ce que vous écrivez dans FRED fonctionnera sur une instance Node-RED fonctionnant sur un Pi ou un Beagleboard.

Pour commencer nos exemples, assurez-vous d'avoir créé votre propre instance Node-RED dans le cloud en vous inscrivant à un compte gratuit sur http://fred.sensetecnic.com .

L'interface utilisateur Node-RED

Une fois que vous vous êtes connecté à votre compte FRED, vous verrez l'interface utilisateur Node-RED standard qui se compose de trois volets principaux, comme illustré à la Fig 2.1

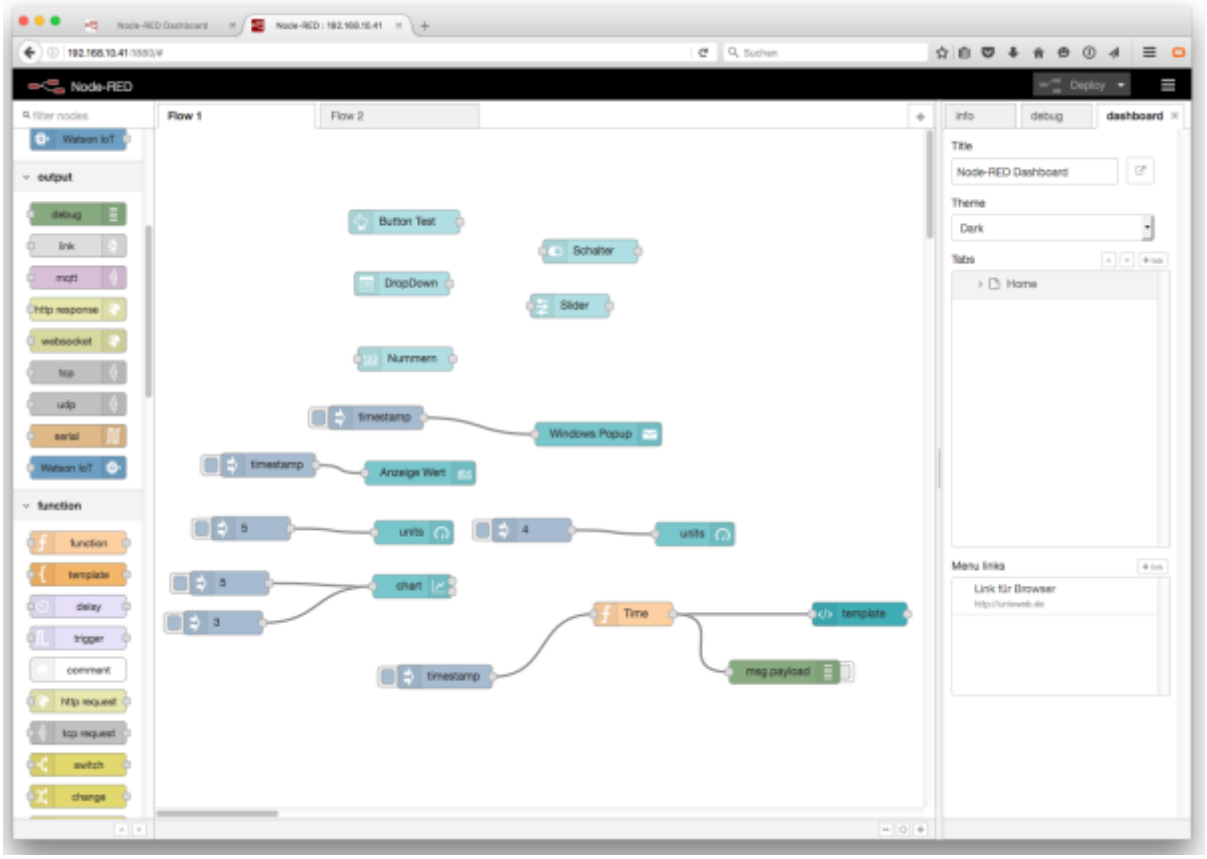


Figure 2.1 L'interface utilisateur Node-RED - montrant la palette de nœuds (à gauche), le canevas de flux (au centre) et le volet de sortie (à droite)

Le volet principal est l'espace de travail de création de flux au milieu. C'est là que vous faites glisser et déposez des nœuds et que vous les connectez avec des fils. En haut du volet de l'espace de travail se trouve un ensemble d'onglets. Chaque onglet ouvre un espace de travail créé précédemment et affiche tous les flux créés à l'aide de cet espace de travail.

Sur la gauche se trouve le volet des nœuds qui contient tous les nœuds intégrés pris en charge par votre instance de Node-RED. Dans l'exemple ci-dessus, vous voyez l'ensemble de nœuds FRED qui contient une grande sélection qui a été ajoutée à l'ensemble de base fourni avec Node-RED. Dans des conférences ultérieures, vous découvrirez ces nœuds et les utiliserez pour développer de nouveaux flux. Comme vous pouvez le voir, les nœuds sont regroupés en catégories. L'ouverture d'une catégorie affiche les nœuds individuels.

Sur le côté droit se trouve le volet de sortie qui peut être basculé entre les onglets d'informations et de débogage. Lorsque info est sélectionné, la documentation du nœud sélectionné y est affichée. Lorsque le débogage est sélectionné, il affiche la sortie des nœuds de débogage, des erreurs et des avertissements.

Au-dessus de ces trois volets principaux se trouve la barre d'outils habituelle, et sur le côté droit se trouvent trois widgets, un bouton de déploiement, une icône d'informations utilisateur et un menu déroulant pour l'administration et le contrôle. Vous examinerez plus en détail le pulldown plus tard dans ces conférences. L'icône d'informations utilisateur vous permet de revenir à la page d'accueil FRED avec des liens vers des tutoriels, vos informations de compte, votre statut et d'autres informations ainsi que de vous déconnecter du service FRED.

Le bouton Déployer est utilisé lorsqu'un flux a été construit et provoque le déploiement du flux sur le système Node-RED et son exécution. Vous découvrirez les détails de ce qui se passe réellement sous les couvertures dans la conférence 5 et au-delà. Pour l'instant, considérez simplement le bouton Déployer comme le moyen de faire fonctionner votre flux.

Un tour rapide des nœuds et des messages Node-RED

Comme vous l'avez vu dans la leçon 1, Node-RED vous permet de câbler ensemble des nœuds pour créer des flux qui exécutent votre tâche de programmation. Les messages passent entre les nœuds, passant des nœuds d'entrée aux nœuds de traitement en passant par les nœuds de sortie. Examinons brièvement les nœuds, les flux et les messages.

Il existe trois principaux types de nœuds :

- Nœuds d'entrée (par exemple, injecter)
- Nœuds de sortie (par exemple, débogage)
- Nœuds de traitement (par exemple, fonction)

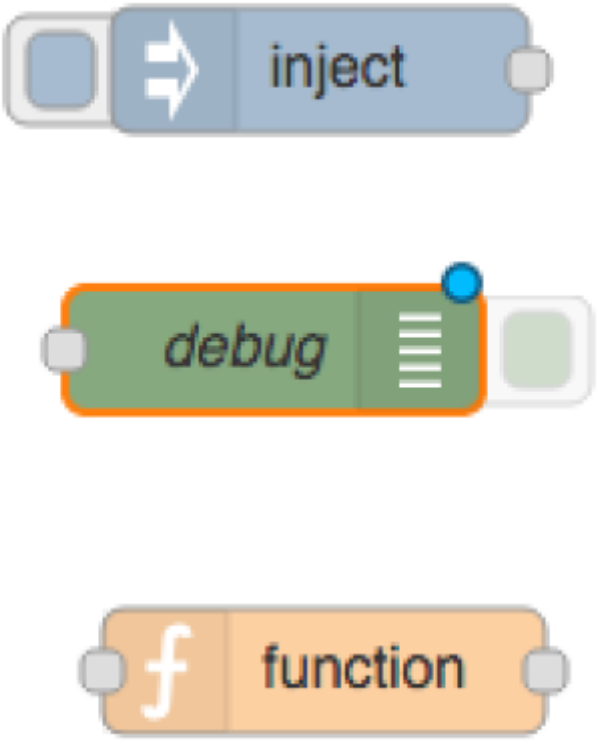


Figure 2.2 Les principaux types de nœuds : entrée, sortie et traitement

Les nœuds d'entrée vous permettent d'entrer des données dans une application ou « flux » Node-RED. Ils ont au moins un point de terminaison de sortie représenté par le petit carré gris uniquement sur leur côté droit. Vous utilisez des nœuds d'entrée pour connecter des données provenant d'autres services, par exemple les nœuds Twitter, Google, série, websockets ou tcp, ou pour saisir

manuellement des données dans un flux à l'aide du nœud d'injection.

Les nœuds de sortie vous permettent d'envoyer des données en dehors d'un flux Node-RED. Ils ont un seul point de terminaison d'entrée sur leur côté gauche. Vous utilisez des nœuds de sortie pour envoyer des données à d'autres services, par exemple via des nœuds Twitter, tcp, série ou e-mail, ou pour utiliser le nœud de débogage pour sortir vers le volet de débogage.

Les nœuds de traitement vous permettent de traiter des données. Ils ont un point de terminaison d'entrée et un ou plusieurs points de terminaison de sortie. Ils vous permettent de transformer les nœuds de type de données (par exemple, json, csv, xml), d'utiliser les données pour déclencher un nœud de message (par exemple, déclencheur, retard) et d'écrire du code personnalisé qui utilise les données reçues (par exemple, un nœud de fonction).

Notez que certains nœuds, comme les messages d'injection et de débogage, ont un bouton qui vous permet d'activer un nœud (dans le cas du nœud d'injection) ou d'activer et de désactiver un nœud (dans le cas du nœud de débogage).

Les flux se composent de plusieurs nœuds câblés ensemble, avec des onglets de sortie liés aux onglets d'entrée du nœud suivant dans le flux. Les messages circulent le long des nœuds transportant des données d'un nœud à l'autre.

Les nœuds Node-RED consomment des messages d'entrée et produisent des messages de sortie. Les messages sont des objets JavaScript qui contiennent au moins un paramètre “payload”, comme ceci :

Listing 2.1 Une structure de message Node-RED de base

codebase.txt

```
msg = {
  payload:”message payload”
};
```

Les nœuds consomment et produisent des messages, en utilisant généralement msg.payload comme espace réservé principal pour les données qu'ils consomment et produisent. Cependant, les messages peuvent être étendus pour contenir d'autres paramètres. Par exemple, pour définir le sujet d'un message et ajouter un nouveau paramètre, l'emplacement, vous pouvez créer un nouvel objet msg comme indiqué dans le listing 2.2.

Listing 2.2 Une structure de message Node-RED plus complexe

codecomplexe001.txt

```
msg = {
  payload:”message payload”,
  topic:”error”,
  location:”somewhere in space and time”
};
```

Utilisons ces connaissances pour créer votre premier flux.

Exemple 2.1 Construire votre premier flux : Hello World

Continuons et commençons à créer votre premier flux afin que vous puissiez voir à quel point il est simple d'utiliser l'interface utilisateur Node-RED pour créer et déployer un flux.

Commençons par le flux le plus simple possible, un nœud pour injecter des informations dans le flux, connecté à un nœud de débogage pour voir la sortie du flux sous forme de message de débogage. Une fois que vous l'aurez exécuté, vous le développerez jusqu'au flux Hello World complet.

Comme c'est la première fois qu'on vous montre comment créer un flux, commençons lentement et expliquons chaque étape avec une capture d'écran. Une fois que vous avez créé votre premier flux et vu à quel point c'est facile, vous verrez principalement une seule image du flux final plutôt que toutes les captures d'écran.

Commençons par le Node le plus simple, le Node de commentaire. Vous le trouverez dans la section des fonctions de la palette de nœuds. Faites glisser et déposez un nœud de commentaire sur l'espace de travail du flux, comme illustré à la figure 2.3.

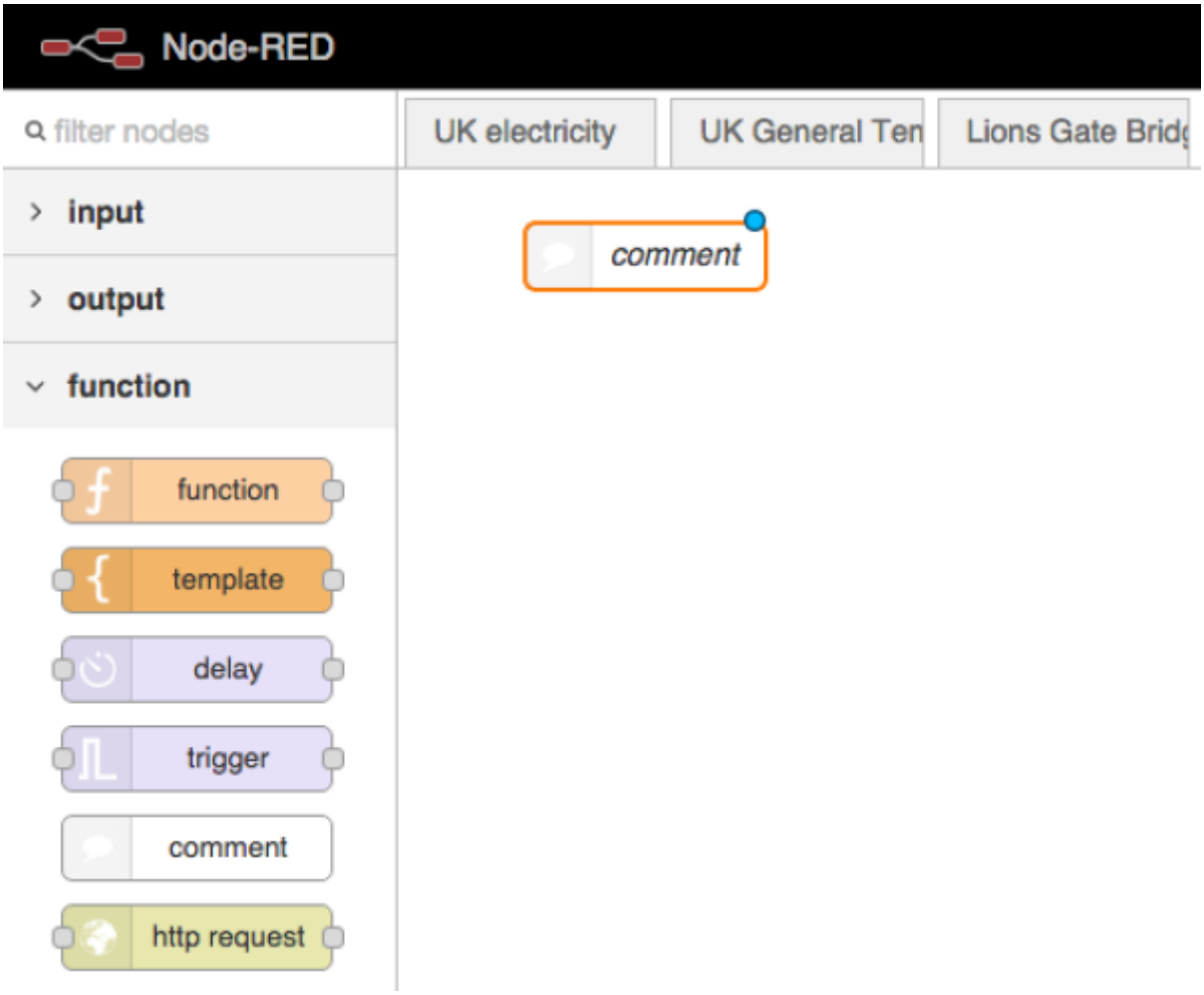


Figure 2.3 : L'utilisation d'un nœud de commentaire est un excellent moyen d'ajouter des commentaires visibles dans les flux

Une fois que vous avez fait cela, jetez un œil au volet d'informations sur la droite (n'oubliez pas de passer aux informations si l'onglet de débogage est sélectionné). Vous verrez quelques informations sur le nœud, y compris le nom du nœud, un identifiant unique et un champ de propriétés avec une description du nœud. Pour le nœud de commentaire, il n'y a pas grand-chose à dire ; cependant, les nœuds plus sophistiqués ont beaucoup plus d'informations.

Double-cliquez sur le nœud de commentaire et vous verrez une fenêtre de configuration (Fig 2.4). Vous pouvez donner un nom au nœud de commentaire et ajouter un texte détaillé si vous le souhaitez.

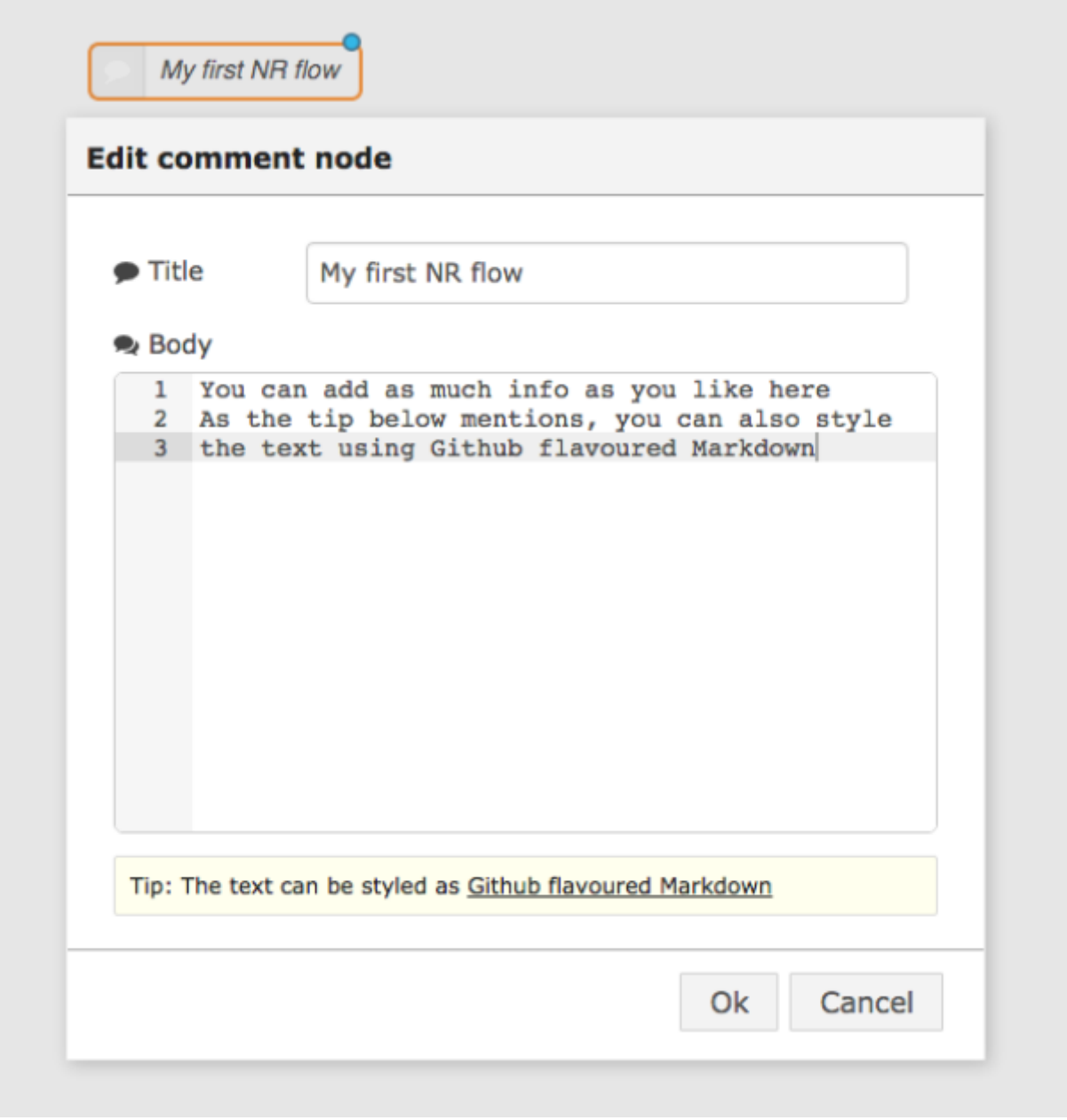


Figure 2.4 Donnez un nom au commentaire et ajoutez les informations souhaitées dans la zone de texte

Ok, ajoutons maintenant le premier nœud qui fait réellement quelque chose, un nœud d'injection. Le nœud d'injection est utilisé pour générer une entrée dans un flux et est l'un des premiers nœuds de la palette de nœuds sous entrée. Si vous faites glisser et déposez un nœud d'injection sur l'espace de travail de flux, puis regardez l'onglet d'informations, vous verrez la documentation du nœud d'injection. Notez que le nom du nœud sur l'espace de travail change d'injection à horodatage, car le comportement par défaut du nœud consiste à injecter un horodatage - l'heure actuelle en millisecondes depuis le 1er janvier 1970.

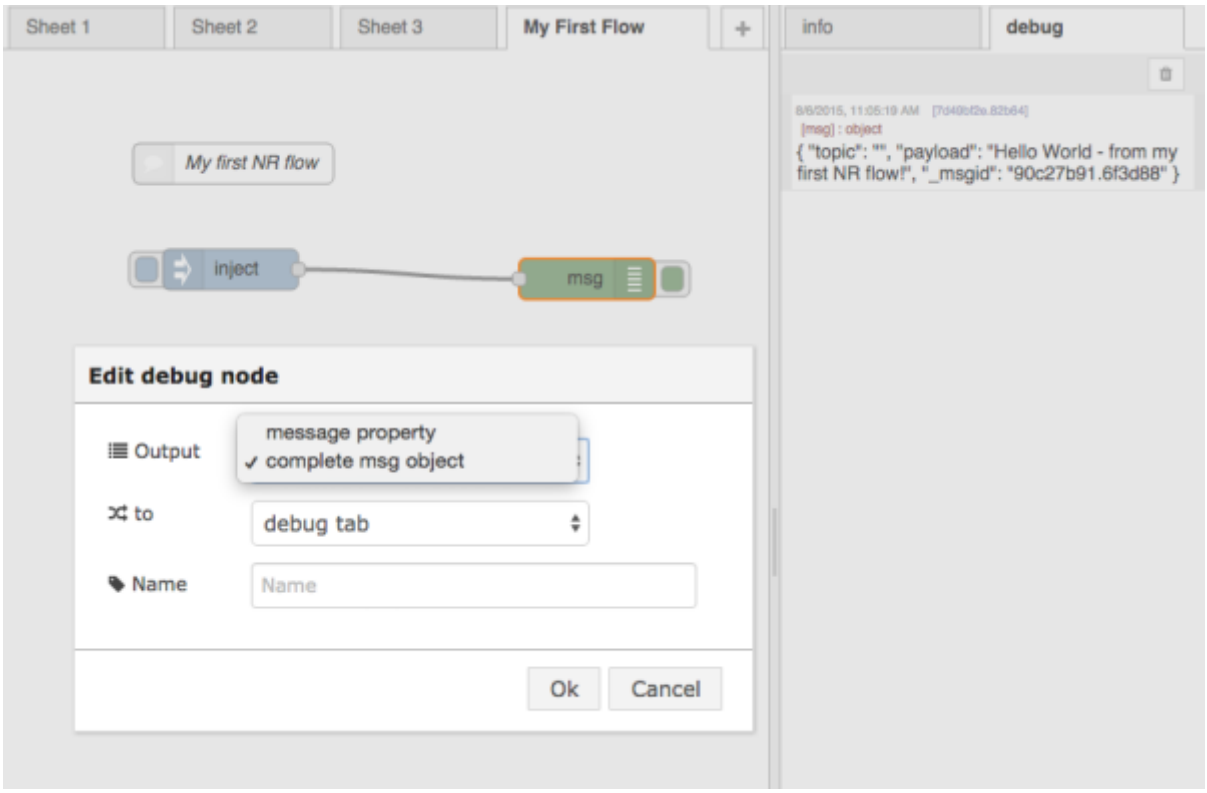


Fig 2.5 Le nœud d'injection vous permet d'insérer des événements sous forme de messages, par défaut un horodatage

Vous remarquerez également que le nœud d'injection (maintenant nommé horodatage) a un point bleu en haut à droite et un carré gris au centre à droite. Le point bleu indique que le nœud n'a pas été déployé depuis sa dernière modification ; le carré gris est le point de sortie du nœud. C'est là que vous attachez des « câbles » qui acheminent le message de sortie du nœud d'injection vers le nœud suivant dans le flux.

Pour avoir une idée du nœud d'injection et de l'ensemble du processus de déploiement du flux, ajoutons un nœud de débogage pour voir ce qui se passe, connectons-les ensemble, puis déployons le flux et testons-le.

Commencez par faire glisser un nœud de débogage de la palette de nœuds vers l'espace de travail. Encore une fois, vous pouvez regarder les informations pour le nœud.

Ensuite, vous connecterez les deux nœuds ensemble. Pour ce faire, cliquez sur le point de sortie gris du nœud d'injection et, en maintenant le bouton de la souris enfoncé, faites glisser vers le nœud de débogage. Un fil orange apparaît, que vous pouvez ensuite attacher au point d'entrée gris sur le nœud de débogage.

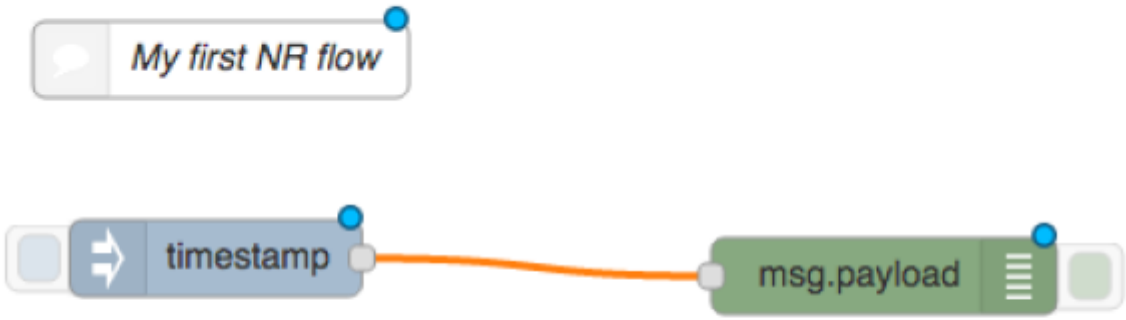


Figure 2.6 Câblage d'un nœud d'injection à un nœud de débogage

Il s'agit du flux le plus simple possible et enverra l'horodatage actuel au nœud de débogage pour l'afficher dans le volet de débogage. Essayons!

Cliquez sur le bouton de déploiement dans la fenêtre Node-RED (en haut à droite). Vous verrez une fenêtre contextuelle indiquant que le flux a été déployé avec succès. Vous remarquerez également que les points bleus sur les nœuds disparaissent, indiquant qu'il n'y a pas de modifications non déployées.

Maintenant, avant d'essayer le flux, assurez-vous que l'onglet de débogage est sélectionné dans le volet de droite. Cliquez ensuite sur l'onglet de gauche du nœud d'injection et regardez ce qui apparaît dans le volet de débogage.

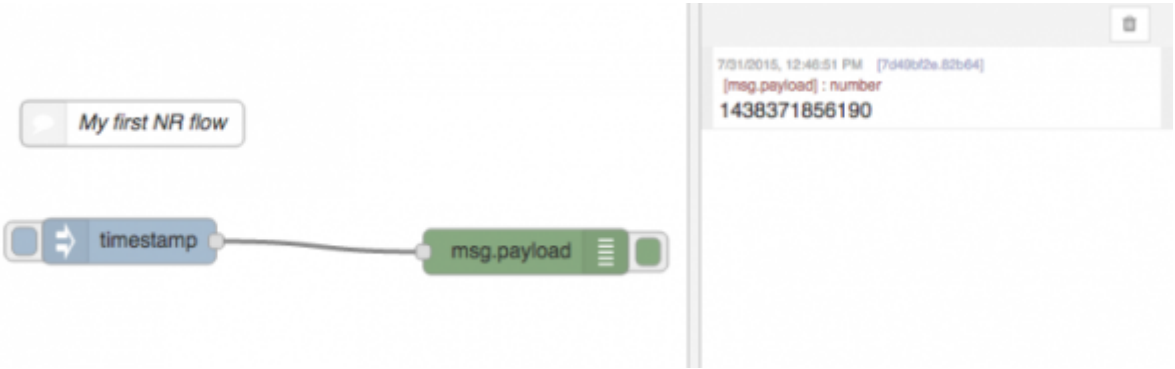


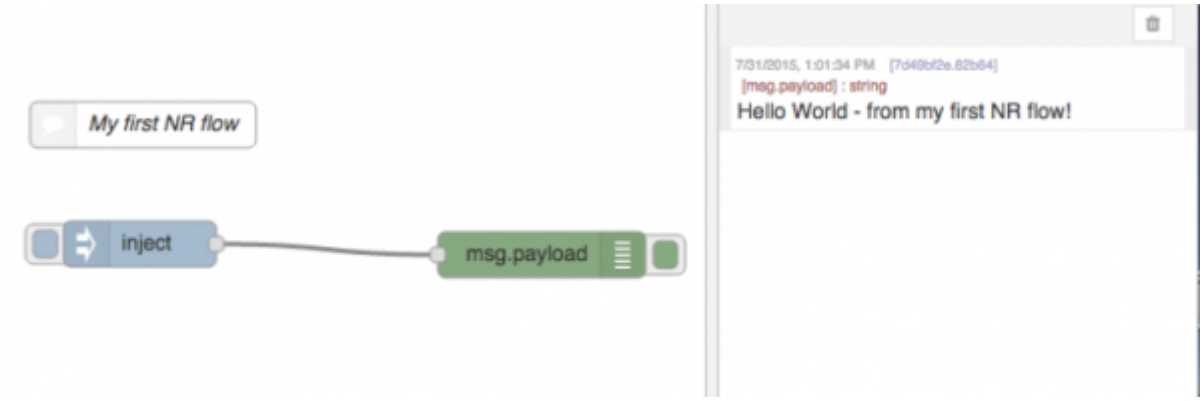
Figure 2.7 Sortie de débogage de votre premier flux - l'heure à laquelle vous avez cliqué sur le nœud d'injection.

Comme vous pouvez le voir, le nœud d'injection, lorsqu'il est cliqué, génère un horodatage (nombre de millisecondes depuis le 1er janvier 1970), qui est converti en un message et envoyé le long du fil

de sortie, qui est transmis au nœud de débogage en tant que message d'entrée. . Le comportement par défaut du nœud de débogage est d'afficher tous les messages qu'il reçoit, ce qu'il fait dans le volet de débogage à droite.

Félicitations, vous avez créé et déployé votre premier flux !

Augmentons-le maintenant un peu pour voir ce que nous pouvons faire d'autre avec ce flux simple. Tout d'abord, nous allons modifier le nœud d'injection pour envoyer un message texte plutôt qu'un horodatage. Pour ce faire, sélectionnez le nœud d'injection dans le flux et double-cliquez dessus. Vous verrez une fenêtre de configuration comme celle de la figure 2.8.



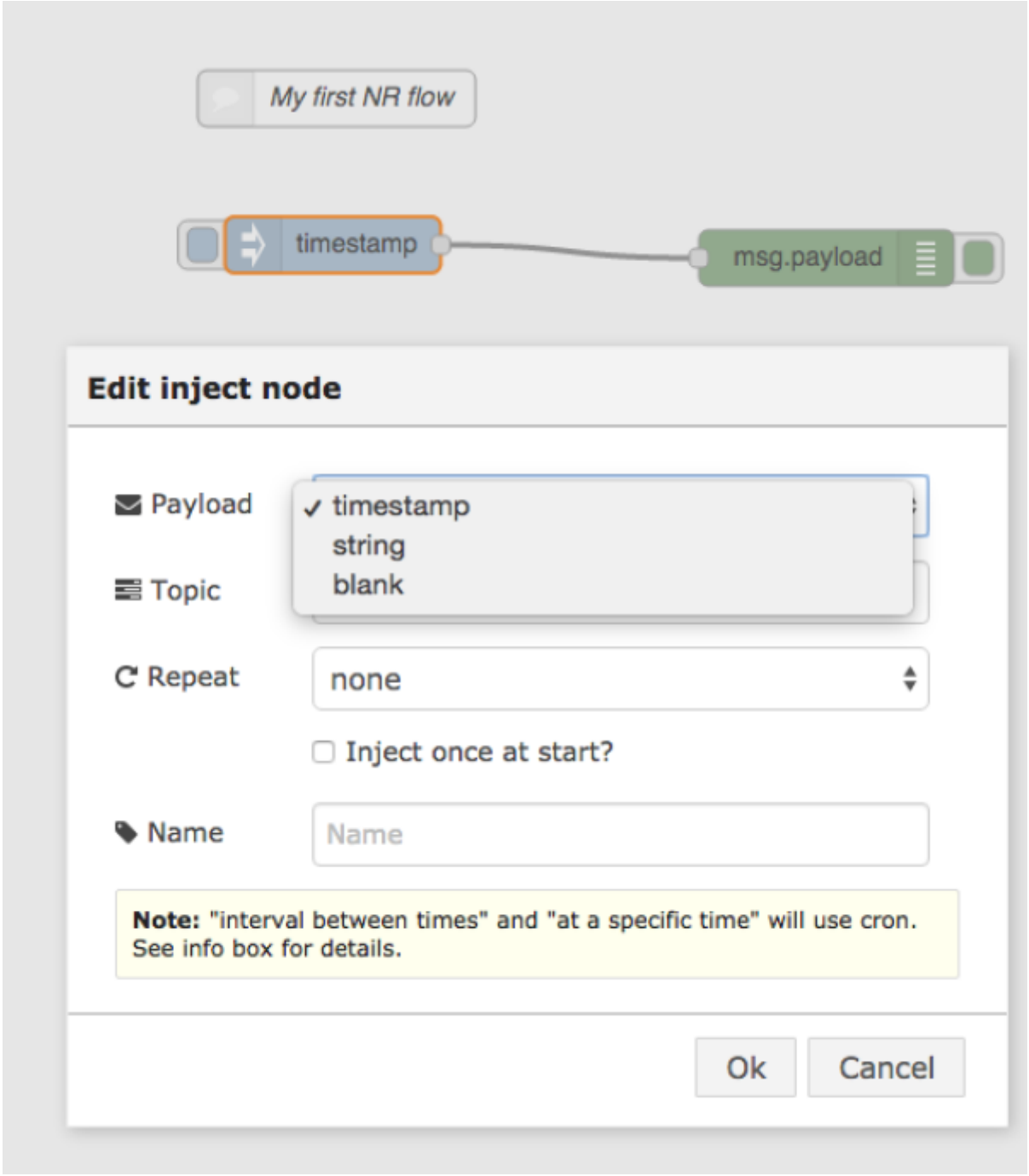


Figure 2.8 Modification d'un nœud d'injection pour envoyer du texte au lieu d'un horodatage

Dans le champ de charge utile, sélectionnez chaîne au lieu de l'horodatage, puis saisissez la chaîne de votre choix dans le champ vide sous le champ de charge utile. Comme le veut la tradition, commençons par “Hello World - from my first NR flow!”

Une fois que vous avez effectué la modification, cliquez sur OK pour enregistrer les modifications et jeter un œil au flux. Vous verrez que le point bleu est apparu sur le nœud d'injection (qui est également redevenu inject plutôt que timestamp) pour indiquer que vous avez des modifications non déployées. Cliquez à nouveau sur le bouton de déploiement pour résoudre ce problème, puis continuez et cliquez sur l'onglet du nœud d'injection. Si vous regardez la sortie de débogage, vous verrez qu'au lieu d'un horodatage, votre texte a été livré sous forme de message au nœud de débogage, qui l'affiche comme d'habitude.

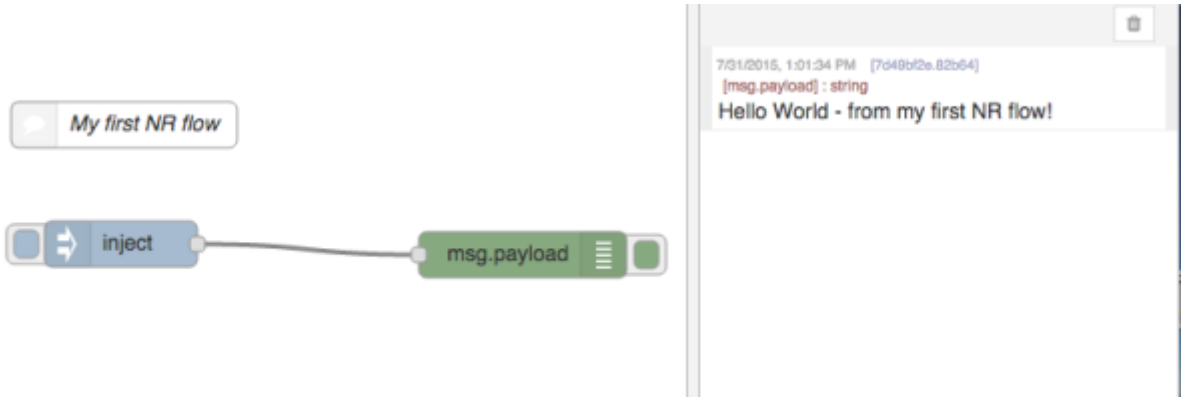


Figure 2.9 Envoi d'un message texte plutôt qu'un horodatage

Comme vous pouvez le constater, il est très facile de connecter des nœuds simples et de faire passer les données dans votre flux sous forme de messages. À ce stade, vous vous poserez probablement des questions sur les messages qui circulent entre les nœuds. Le nœud de débogage peut être utilisé pour les examiner plus en détail.

Tous les messages dans Node-RED ont trois propriétés par défaut : la charge utile, que nous voyons ci-dessus, un sujet de message, qui est une chaîne définie par l'utilisateur décrivant le sujet du message (son sujet, si vous voulez) et un identifiant interne. Vous pouvez réellement voir ces informations si vous modifiez la configuration du nœud de débogage. Faisons cela et regardons la sortie (Fig 2.10)

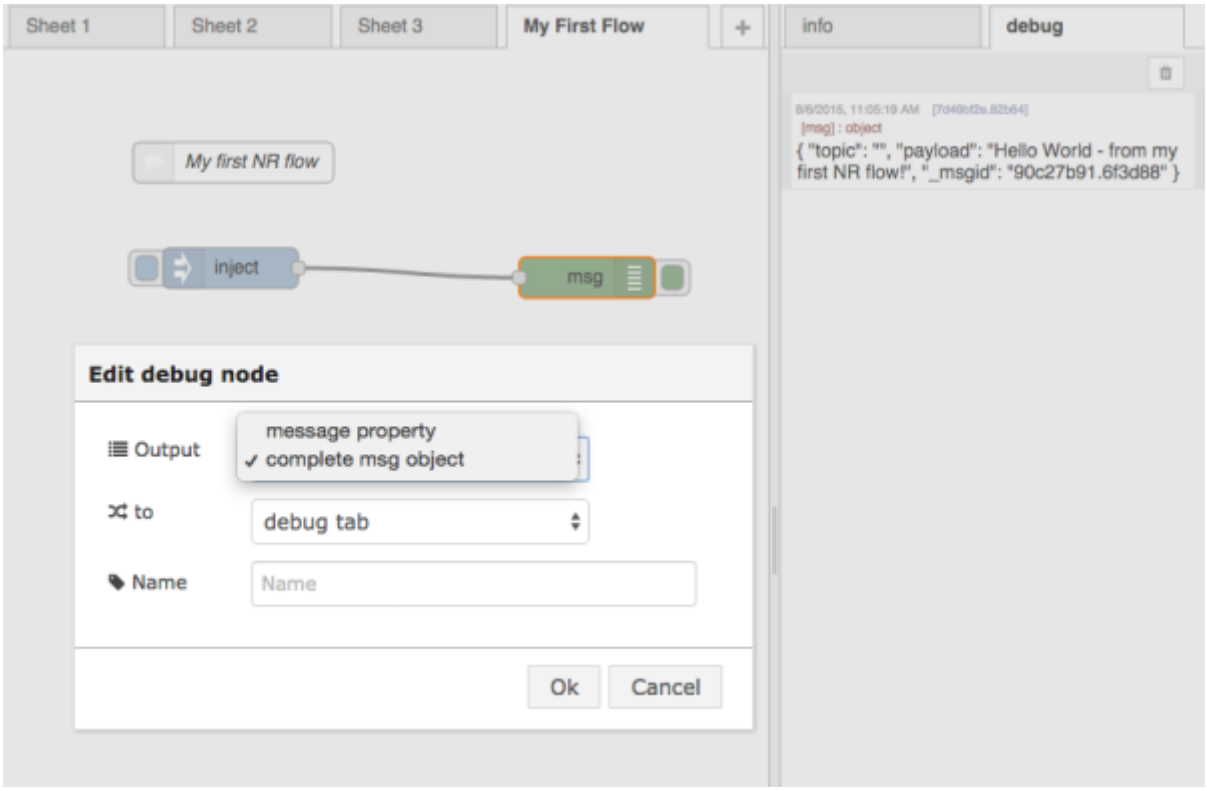


Figure 2.10 : configuration du nœud de débogage pour afficher les éléments internes d'un message

From:
http://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
http://chanterie37.fr/fablab37110/doku.php?id=start:rasberry:nodered:doc&rev=1679690039

Last update: **2023/03/24 21:33**