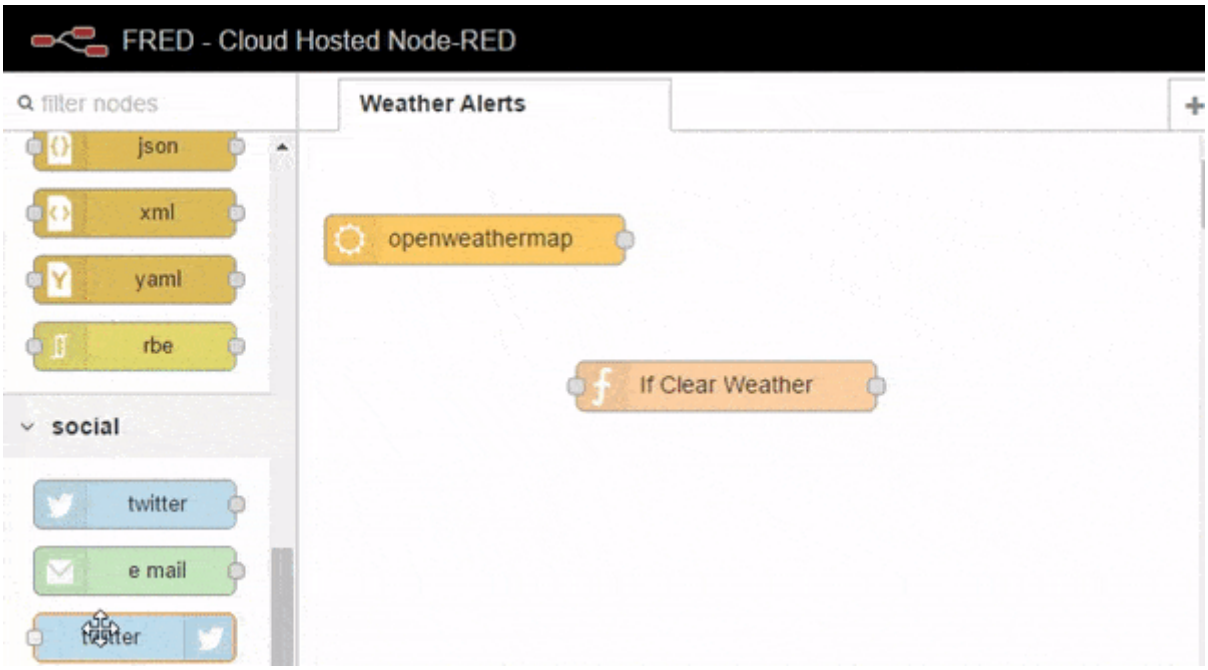
بناء التدفقات

هذا الدرس سيتم عرض واجهة node-RED و نبدأ في بناء التدفق الأول. عليك أن تتعلم كيفية إنشاء تدفقات بسيطة، و استخدام عقد debug لتعقب الرسائل خلال التدفقات و كيفية استخدام عقدة function لكتابة كود جافا سكريبت JavaScript بسيط لتقوم العقدة بتلبية احتياجاتك المحددة .



في نهاية هذا الدرس ستكون قادر على انشاء تدفقات بسيطة خاصة بك، عليك أن تتعرف على واجهة المستخدم للـ node-RED و عليك أن تكون على دراية ببعض العقد المضمنة في Node-RED .

للبدأ، عليك استخدام cloude service التي توفر لك Node-RED ومجموعة واسعة من العقد المبنية.

FRED

FRED هو خدمة Cloude التي تم انشاءها لتسمح لك للتسجيل و استخدام Node-RED من دون تنزيله و تثبيته على جهازك مثل الراسبيري باي. وكما انها تشمل مجموعة من العقد الإضافية التي ساهمت في اضافتها فئات من المجتمع و التي يمكنك إضافتها إلى المجموعة الأساسيى الموجودة في Node-RED . ومع ذلك، FRED هو مجرد غلاف لـ Node-RED ، وبالتالي فإن أي شيء تكتبه في FRED سيعمل على الـ Node-RED بشكل مشابه للعمل على جهاز اخر كالراسبيري باي.

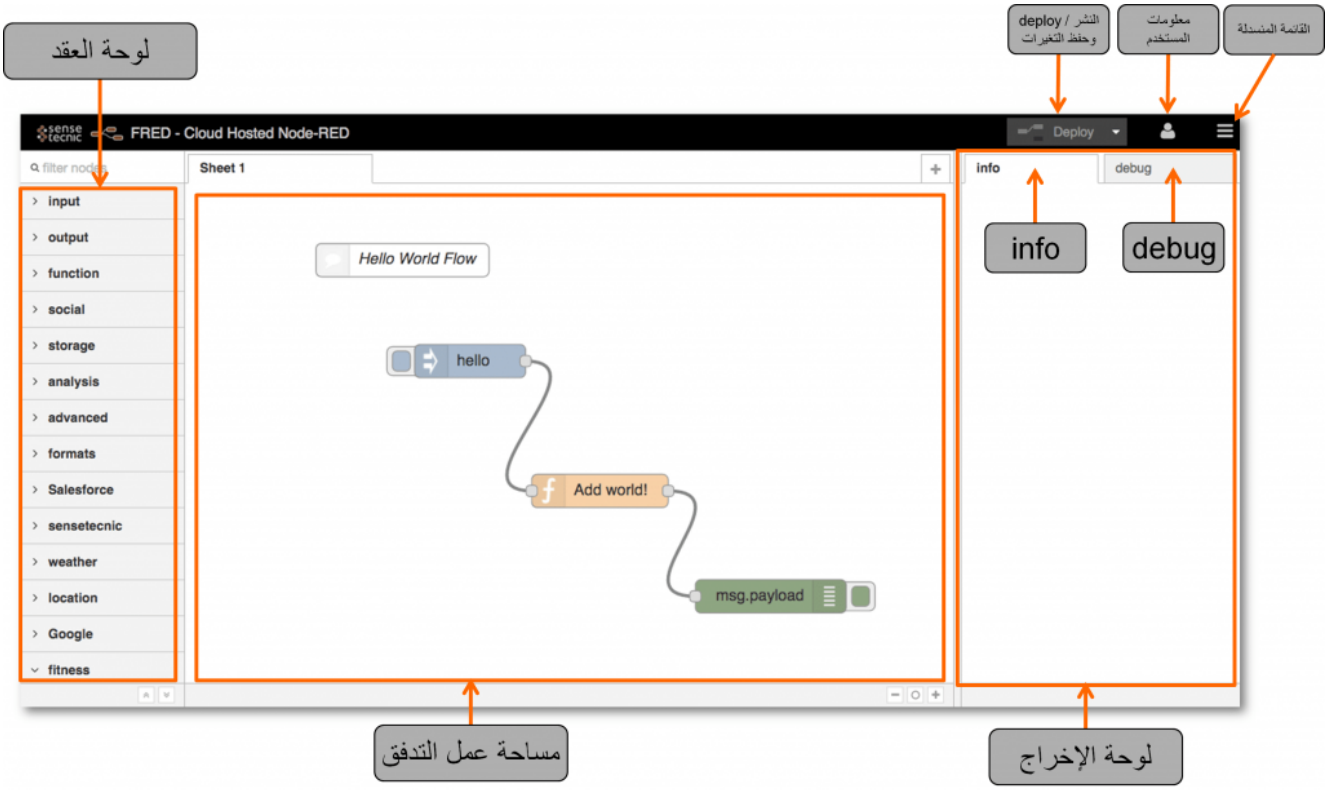
لبدء مثالنا تأكد من أنك قمت بإنشاء Node-RED الخاص بك على Cloud بواسطت انشاء حساب مجاني في
http://fred.sensetecnic.com

بعد التسجيل، تأكد من تفعيل الحساب الخاص بك عبر البريد الإلكتروني الخاص بك. فلن تكون قادر على تسجيل الدخول حتى يتم

التحقق من صحة الحساب.

واجهة المستخدم للـ Node-RED

بمجرد دخولك إلى حساب FRED، سترى واجهة المستخدم للـ Node-RED التي تتكون من ثلاثة أجزاء رئيسية ، كما هو مبين بالشكل أدناه .



الجزء الرئيسي هو مساحة العمل لإنشاء التدفق في الوسط. هذا هو المكان الذي يتم سحب و إسقاط العقد و ربطها بالاسلاك عليه. على طول الجزء العلوي من جزء مساحة العمل يوجد مجموعة من علامات التويب . كل تبويب يفتح مساحة العمل التي تم إنشاؤها مسبقا و يظهر التدفقات التي تم إنشاؤها باستخدام مساحة العمل تلك.

على اليسار هو لوحة العقد و التي تحتوي على كافة العقد المدمجة. في المثال المبين في الشكل ، ترى مجموعة عقدة FRED و التي تحتوي على المجموعة التي تم إضافتها إلى المجموعة الأساسية التي تأتي مع Node-RED . في الدروس القادمة سوف تتعلم المزيد عن هذة العقدة و استخدامها لتطوير تدفقات جديدة. كما ترون يتم تجميع العقد إلى فئات . يتم فتح كل فئة لتحصل على العقد .

على الجانب الأيمن هو لوحة الإخراج/الانتاج ، يوجد علامتين تبويب للوحة الإخراج هما info و debug .عندما يتم اختيار info ، يتم عرض الوثائق الخاصة للعقدة المحددة . و عندما يتم اختيار debug، فإنه سيتم عرض المخرجات لعقد الـ debug ، و الأخطاء و التحذيرات.

وفوق هذة الأجزاء الثلاثة الرئيسية هي الأدوات المعتادة،و على الجانب الأيمن ثلاث ازرار، زر النشر (deploy button)، و رمز معلومات المستخدم ، وقائمة منسدلة للإشراف و التحكم. سوف نتطلع على المزيد من التفاصيل في القائمة المنسدلة في وقت لاحق من هذا الدرس. رمز معلومات المستخدم يسمح لك بالعودة إلى صفحة FRED ، و معلومات حسابك ، و غيرها من المعلومات و كذلك لتسجيل الخروج من الخدمة FRED .

يستخدم زر النشر (deploy button) عندما يتم بناء التدفق فهو يسبب نشر للتدفق على نظام Node-RED وتنفيذها. سيتم الدخول لتفاصيل ما يحدث في الواقع في الدروس الاحقة ، الأن زر النشر يمثل طريقة للحصول على التدفق .

جولة سريعة حول عقد الـ Node-RED و تدفق الرسائل

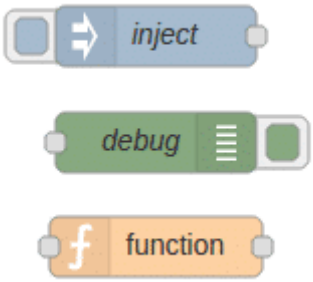
كما عرض عليك بالسابق، الـ Node-RED تتيح لك ربط العقد ببعضها البعض لإنشاء تدفقات ليتم تنفيذ المهام. الرسائل تمر بين العقد، فتتحرك من عقدة الإدخال مرورا بعقدة المعالجة إلى عقدة الإخراج.

يوجد هناك ثلاثة انواع من العقد الرئيسية:

1- عقد الإدخال (مثل : .(inject

2- عقد الإخراج (مثل : .(debug

3- عقد المعالجة (مثل : function) .



عقد الإدخال تسمح لك بإدخال البيانات إلى تطبيق الـ Node-RED او التدفق "flow" . ويوجد لديهم على الأقل نقطة إخراج واحدة يتم تمثيلها بمربع صغير رمادي على الجانب الأيمن للعقدة. ستقوم بإستخدام عقد الإدخال لجلب البيانات من خدمات أخرى مثل عقد التويتر ، serial ، google و tcp. أو ادخال البيانات يدويا إلى التدفق بإستخدام عقدة inject .

عقد الإخراج تتيح لك لإخراج البيانات خارج تدفق الـ Node-RED . فهم يحتوا على نقطة ادخال منفردة على الجانب الأيسر للعقدة. يمكنك استخدام عقد الإخراج لإرسال البيانات إلى خدمات أخرى ، مثل عقد tcp ، twitter ، و عقد الايميل ، أو يمكن استخدام عقدة debug لإخراج المخراج على لوحة الاخراج الـ debug .

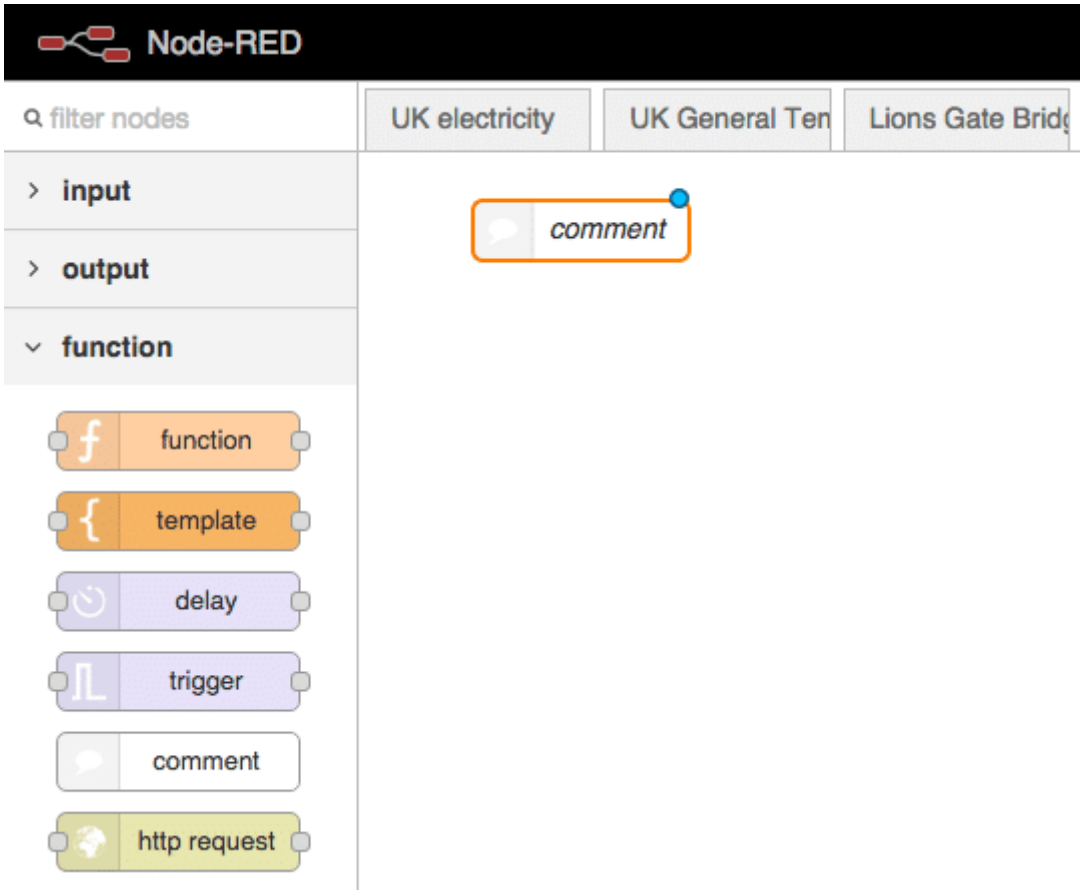
عقد المعالجة تسمح لك بمعالجة البيانات. ولديهم نقطة ادخال و واحد أو أكثر من نقاط الإخراج. حيث انها تتيح لك نقل البيانات مثل عقد (json, csv, xml)، و استخدام البيانات لتحريك الرسائل مثل عقد (trigger, delay) ، ولكتابة التعليمات البرمجية المخصصة لإستخدام البيانات الواردة مثل عقد (function) .

لاحظ أن بعض العقد، مثل inject و debug ، تحتوي على زر يسمح لك بتفعيل العقدة (inject) أو لتمكين و تعطيل العقدة (debug) .

بناء التدفق الأول (Hello World):

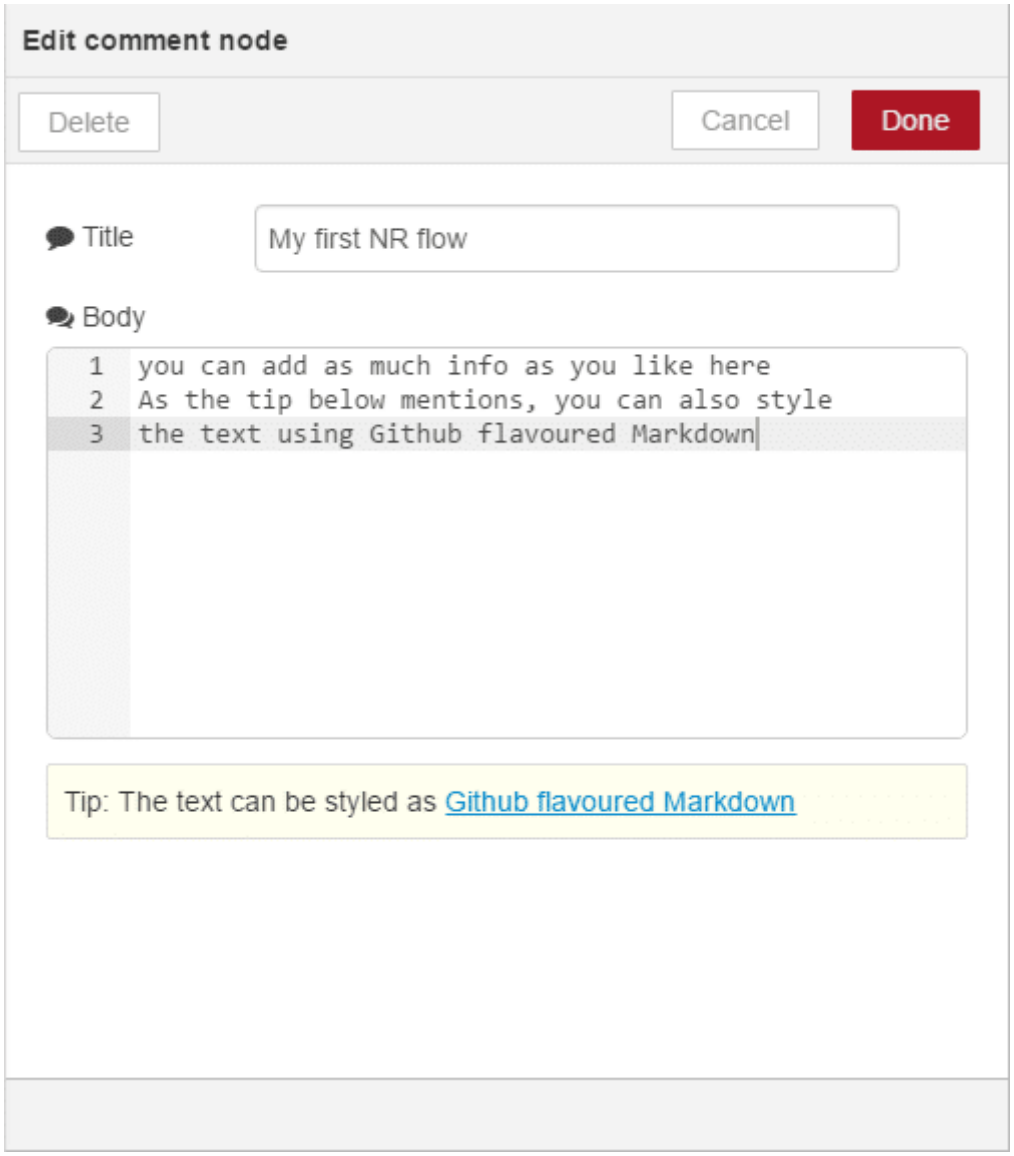
سنقوم في البدء لبناء التدفق الأول الخاص بك بحيث يمكنك أن ترى كم هو بسيط استخدام واجهة مستخدم الـ Node-RED لبناء و نشر التدفق.

دعونا نبدأ من أسهل تدفق محتمل ، سيتم استخدام عقدة لضخ المعلومات في التدفق تدعى (inject node) ، يتم توصيل هذة العقدة إلى عقدة debug لرؤية الناتج من التدفق كرسالة .

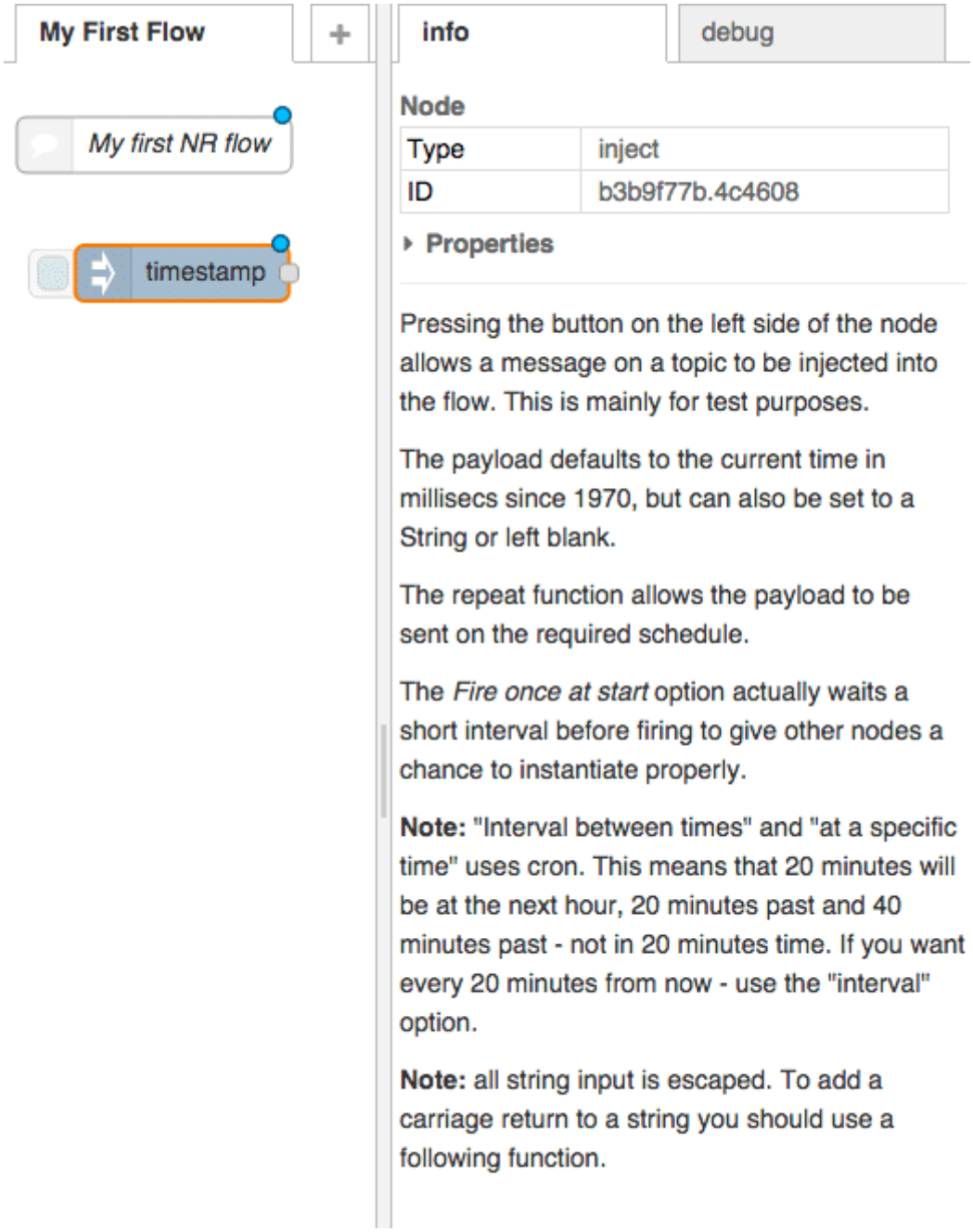


بعد وضع عقدة التعليق (comment Node)على مساحة العمل ، نلقي نظرة على لوحة الخراج عند تويب (info)المعلومات في الجانب الأيمن (قم بالتحويل إلى تبويب info اذا كانت علامة التويب debug) . سترى القليل من المعلومات حول العقدة ، بما في ذلك اسم/ نوع العقدة ، و unique ID ، و حقل الخصائص مع وصفا للعقدة. بالنسبه لعقدة التعليق ليس لديها الكثير ليتم وصفه.

انقر نقرا مزدوجا فوق عقدة التعليق، وسيظهر لك نافذة الإعداد كما في الشكل التالي ، يمكنك إعطاء عقدة التعليق اسم و إضافة نص منفصل إذا أردت.

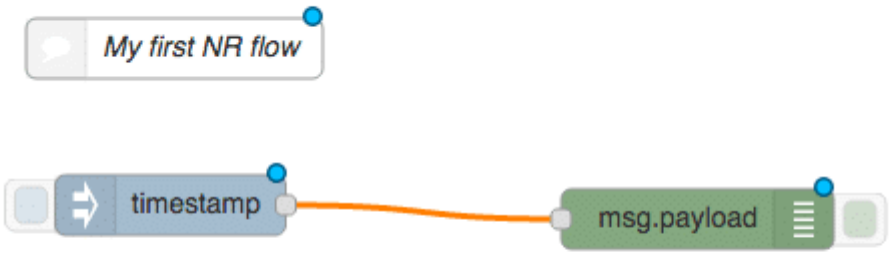


الآن سنقوم بإضافة العقدة الأولى عقدة inject . عقدة inject يتم استخدامها لتوليد الدخل في التدفق وهي العقدة الأولى من العقد في لوحة العقد تحت قسم الإدخال (input) . قم بسحب و إسقاط عقدة inject على مساحة عمل التدفق ، ثم قم بالنظر إلى علامة التبويب معلومات (info) ، سترى معلومات ووصف للعقدة inject . لاحظ ان اسم العقدة يتغير عند وضعها على مساحة العمل من inject إلى timestamp .

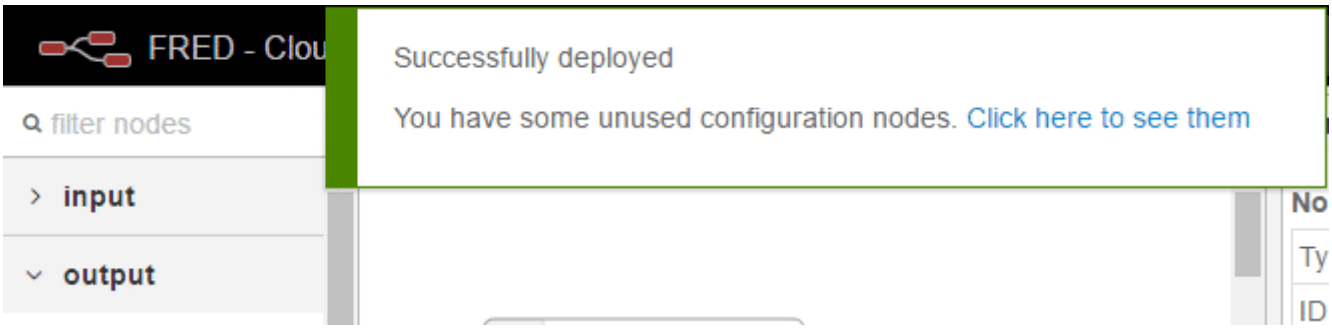


ستلاحظ أيضا أن عقدة inject التي تسمى الأن (timestamp) لديها نقطة زرقاء أعلى اليمين ومربع رمادي في وسط الجانب الأيمن . تشير النقطة الزرقاء أن العقدة لم يتم نشرها لأنه طرأ تغيير عليها. و المربع الرمادي هو نقطة إنتاج/إخراج العقدة. ومن هذا المربع يتم ربط عقدة inject ليتم تدفق الرسالة المنتجة إلى العقدة التالية بالتدفق .

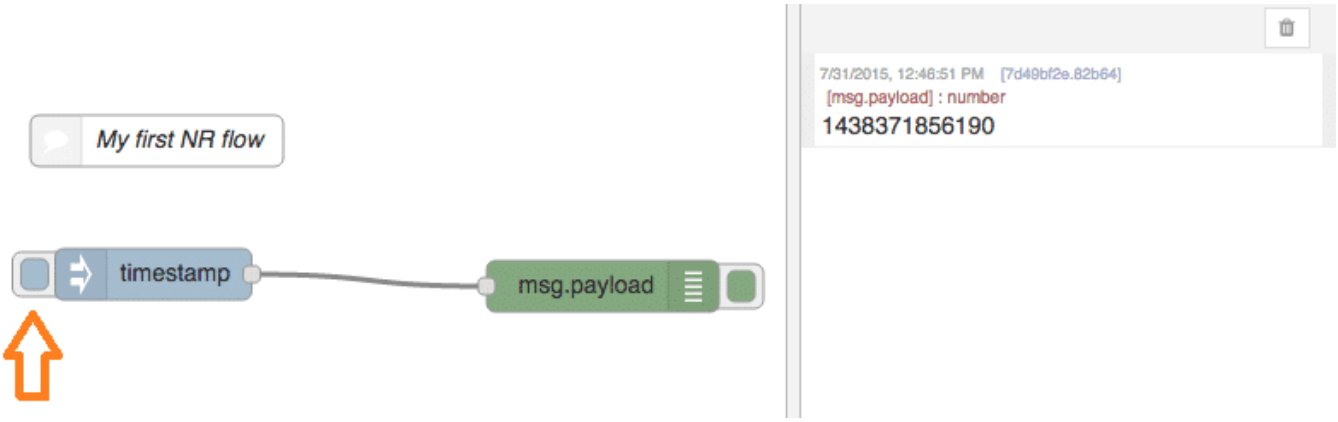
دعونا نضيف عقدة debug سنقوم بربط هذة العقدة مع العقدة السابقة ، وبعد ذلك نقوم بنشر التدفق و اختباره.
قم بسحب عقدة debug من لوحة العقد تحت قسم المخراجات (output) إلى ساحة العمل . مرة أخرى يمكنك الإطلاع على معلومات العقدة في الجزأ الأيمن للواجهة.
ثم قم بربط عقدة debug مع العقدة السابقة معا. للقيام بذلك، انقر على نقطة الإنتاج الرمادية لعقدة inject ، ثم قم بسحب السلك لعقدة debug .



هذا هو ابسط تدفق ممكن، سوف يتم ارسال الزمن الحالي إلى عقدة debug لعرضه على لوحة debug . قم بالضغط على زر النشر deploy في نافذة Node-RED أعلى اليمين ، ستظهر لك نافذة تخبرك أنه تم نشر التدفق بنجاح (The Flow has been successfully deployed) كما هو موضح بالصورة ادناه . و ستلاحظ أن النقطة الزرقاء أعلى يمين العقد لم تعد تظهر.

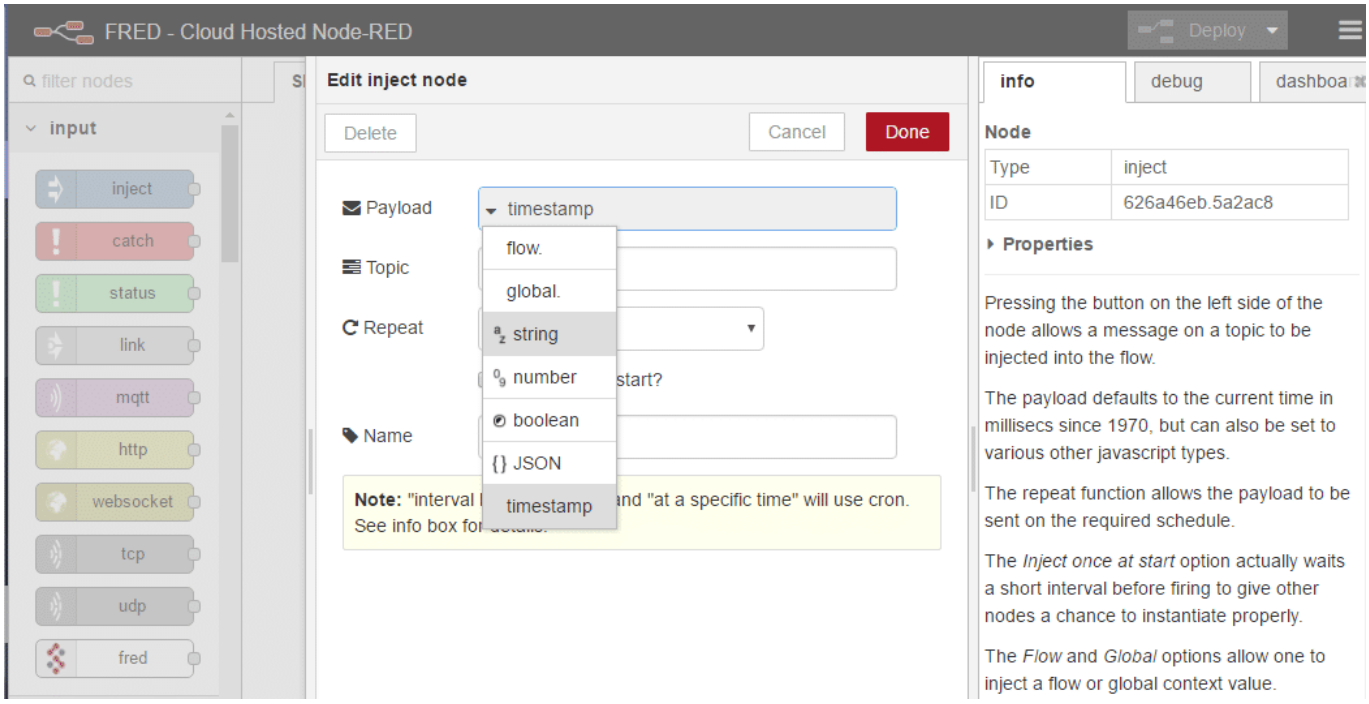


قبل أن يتم اختبار التدفق ، تأكد من اختيار علامة التبويب debug على يمين النافذة . ثم قم بنقر زر التبويب على الجزأ الأيسر لعقدة inject ثم لاحظ ماذا سيظهر على لوحة debug يمن النافذة.



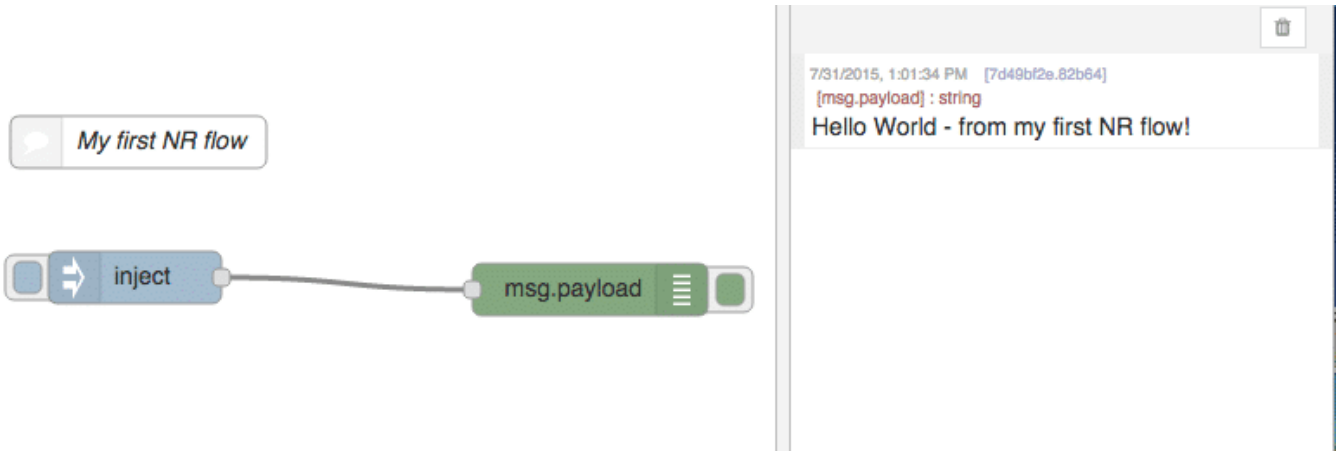
كما ترون، عقدة inject ، عند النقر عليها تولد الطابع الزمني (timestamp) (عدد المللي ثانية) ، والتي يتم تحويلها إلى رسالة و ارسالها على طول سلك الإخراج، ثم يتم تسليمها إلى عقدة debug كرسالة مدخلة. السلوك الإفتراضي لعقدة debug هي إظهار أي رسالة تتلقاها.

الآن سنقوم بإظهار رسالة نصية من خلال هذا التدفق. أولا ، سنقوم بالتعديل على عقدة inject لتقوم بتسليم رسالة نصية بدلا من الطابع الزمني. للقيام بذلك، حدد عقدة inject في التدفق و انقر نقرا مزدوجا عليها . سوف تظهر لك نافذة كما هو موضح لك بالصورة التالية.

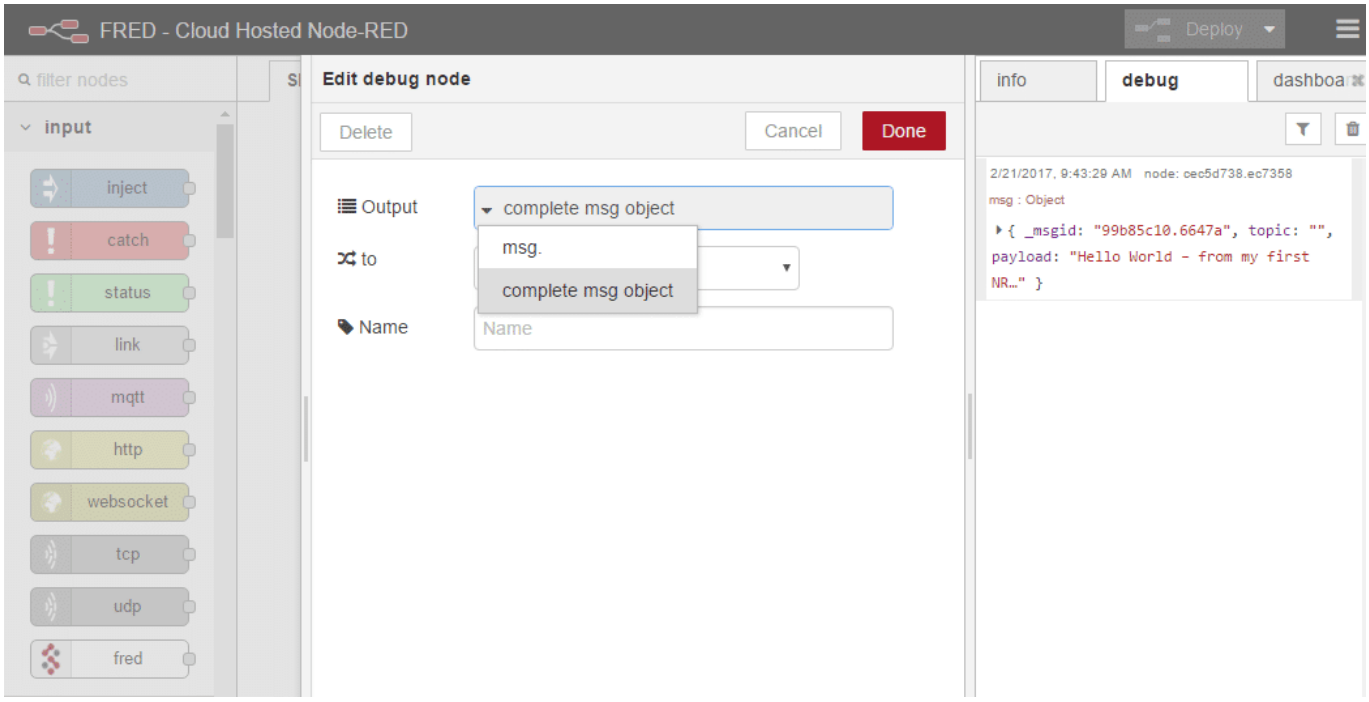


في مربع نص payload ، قم بتحديد string بدلا من الطابع الزمني (timestamp) ثم قم بكتابة اي جملة تريد داخل الحقل الفارغ حقل payload . لنبدأ مع جملة "Hello World – from my first NR flow!"

بمجرد قيامك بهذا التغيير، انقر فوق موافق (Done) لحفظ التغييرات . لنلقي نظرة على التدفق، سترى ظهور النقطة الزرقاء على عقدة inject للإشارة إلى حدوث تغييرات في التدفق لم يتم نشرها. قم بالنقر على زر deploy للنشر ثم قم بالنقر فوق علامة التبويب على عقدة inject على يسار العقدة. إذا قمت بالنظر على لوحة الإخراج debug سترى بدلا من الطابع الزمني سيتم عرض الرسالة النصية.



كما ترون، فإنه من السهل جدا ربط العقد معا و تمرير البيانات من خلال التدفق الخاص بك كرسائل. جميع الرسائل في Node-RED لها ثلاث خصائص افتراضية: payload وهو كما تم عرضها سابقا ، topic وهو موضوع الرسالة لوصف الرسالة المتدفقه ، والمعرف الداخلي. يمكنك أن ترى هذة المعلومات إذا قمت بالتعديل على عقدة debug ، كما هو موضح بالصورة أدناه .



حدد عقدة debug ثم قم بالنقر مرتيين وقم بتغير حقل output إلى " Complete msg object " ثم قم بحفظ التغير، قم بنشر التغير عبر زر deploy ثم قم بالضغط على عقدة inject ، سترى في لوحة الإخراج debug هيكل JSON و الذي يحتوي على ثلاث مجالات : "topic" وهو فارغ حاليا ، و payload و التي تحتوي على جملة الرسالة التي وضعتها في عقدة inject و "ID _msgid" . عموما لا يستخدم ID داخل الرسالة.

التدفق الثاني : تنبيهات الطقس

في هذا المثال ، سيتم مراقبة الطقس لمدينتك و إرسال لنفسك تغريدة في حال يبدو الطقس جيد. عليك استخدام عقدة الطقس (openweathermap) والتي تسترد الطقس من openweathermap.org حول الموقع الذي تم تحديده من قبلك. و سوف يتم استخدام عقدة function للتحقق من الطقس ما اذا كان جيد و عقدة التويتر سيتم استخدامها لإرسال لنفسك تغريدة في حال كان الطقس جيد.

أولا، سوف تحتاج إلى الحصول على مفتاح API في OperWeatherMap . تقدم penWeatherMap معلومات تفصيلية حول الطقس لمواقع في جميع أنحاء العالم . قم بزيارة http://openweathermap.org/appid واتبع المعلومات ادناه .

OpenWeatherMap Home Weather Maps API Price Stations News About

How to work with API key

Home / API / How to work with API key

To get access to weather API you need an API key whatever account you chose from Free to Enterprise.

How to get API key (APPID)

1 Register on the Sign up page

2 Get unique API key on your personal page

How to use API key in API call

Description:

To get access to weather API you need an API key whatever account you chose from Free to Enterprise.

We keep right to not to process API requests without API key.

API call:

http://api.openweathermap.org/data/2.5/forecast/city?id=524901&APPID={APIKEY}

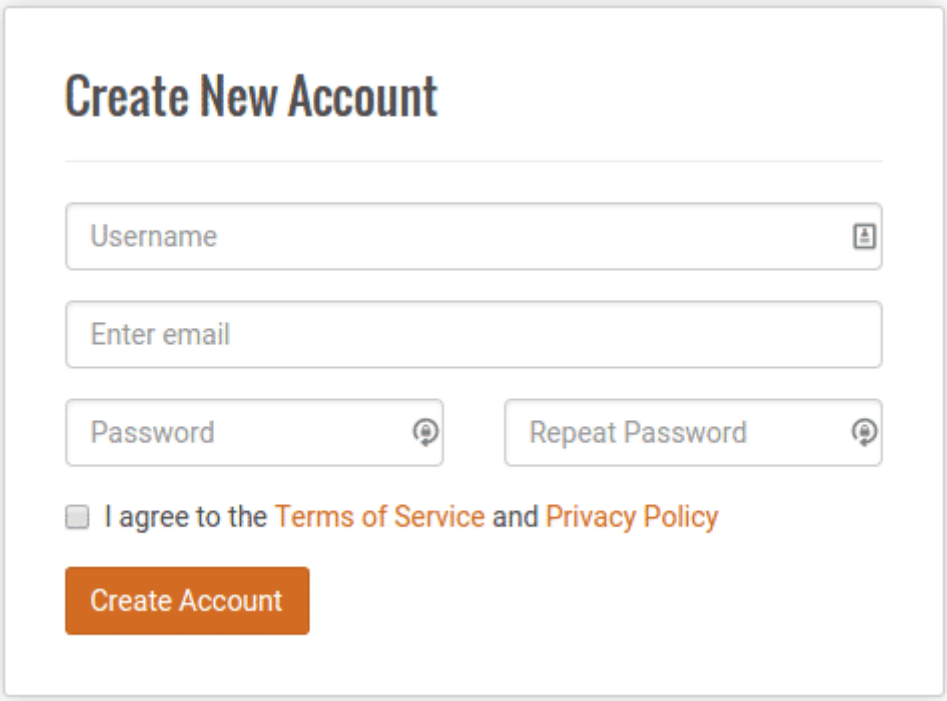
Parameters:

APPID {APIKEY} is your unique API key

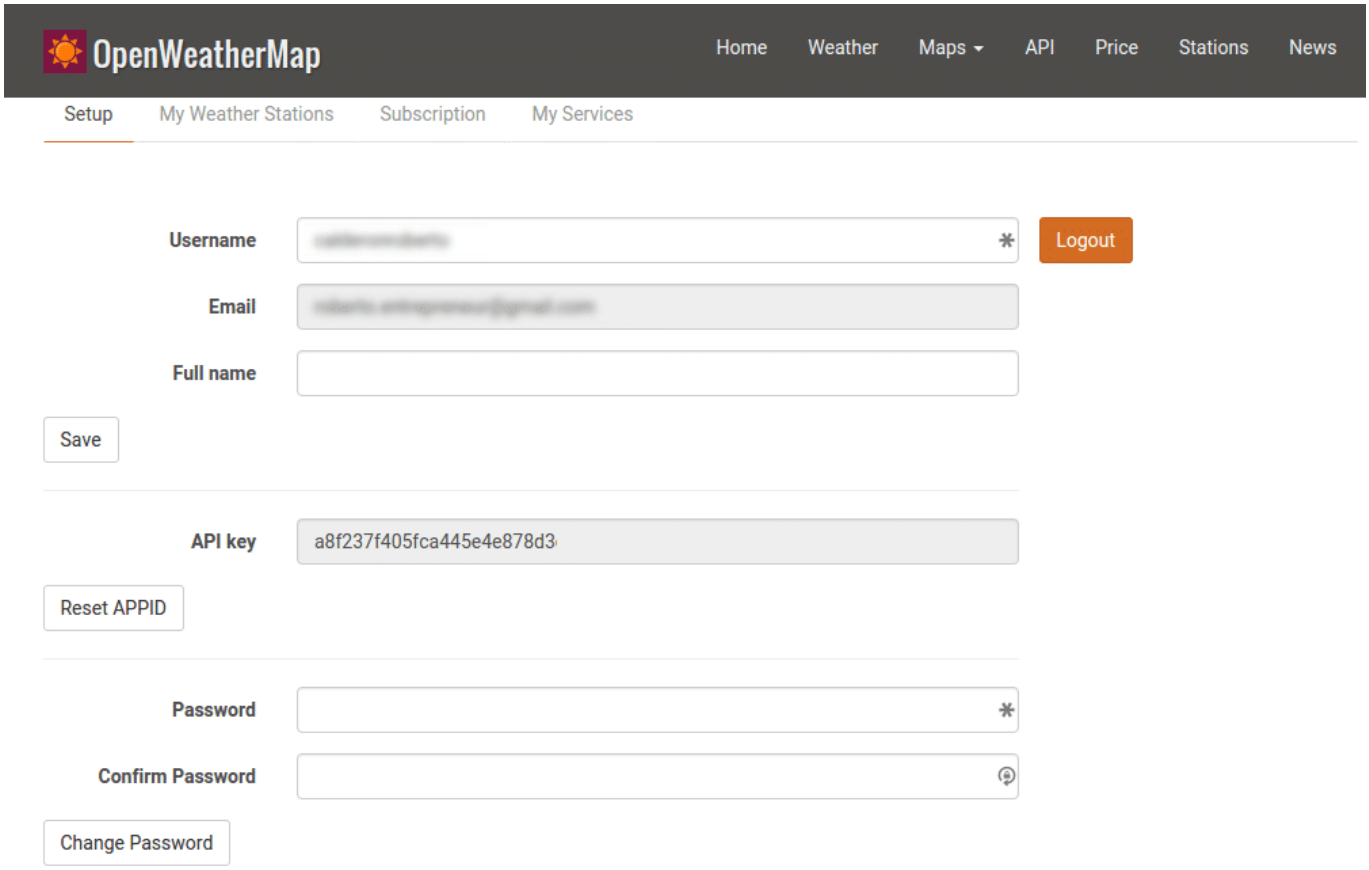
Example of API call:

api.openweathermap.org/data/2.5/forecast/city?id=524901&APPID=1111111111

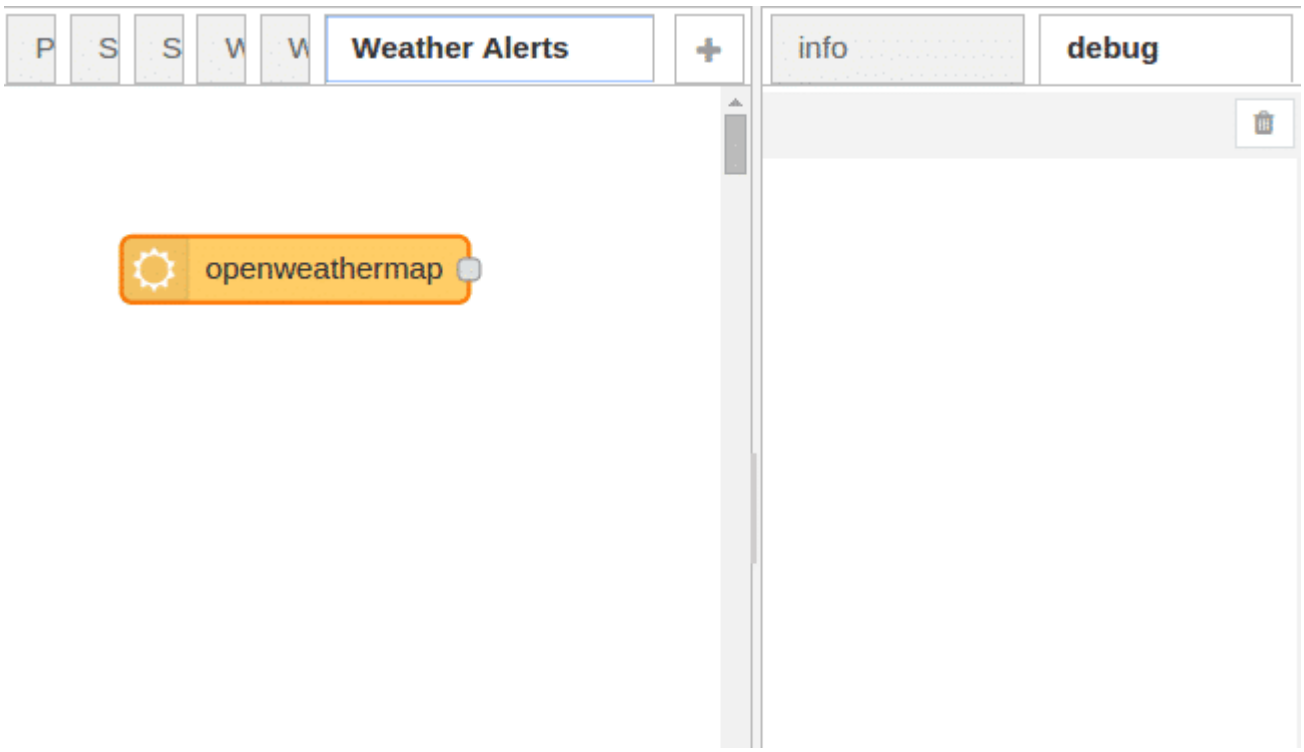
سوف تحتاج إلى الاشتراك للحصول على حساب OpenWeather كما هو موضح ادناه :



بعد الإشتراك سيتم توجيهك إلى الصفحة الرئيسية الخاصة بك حيث أنك ستكون قادرا على الوصول أو إعادة توليد مفتاح API خاص بك .



الآن ، دعونا نستخدم مفتاح API لبناء تدفق تنبيهات الطقس. قم بسحب و اسقاط عقدة الطقس من الجزء الأيمن على مساحة العمل ، كما هو مبين بالشكل ادناه :



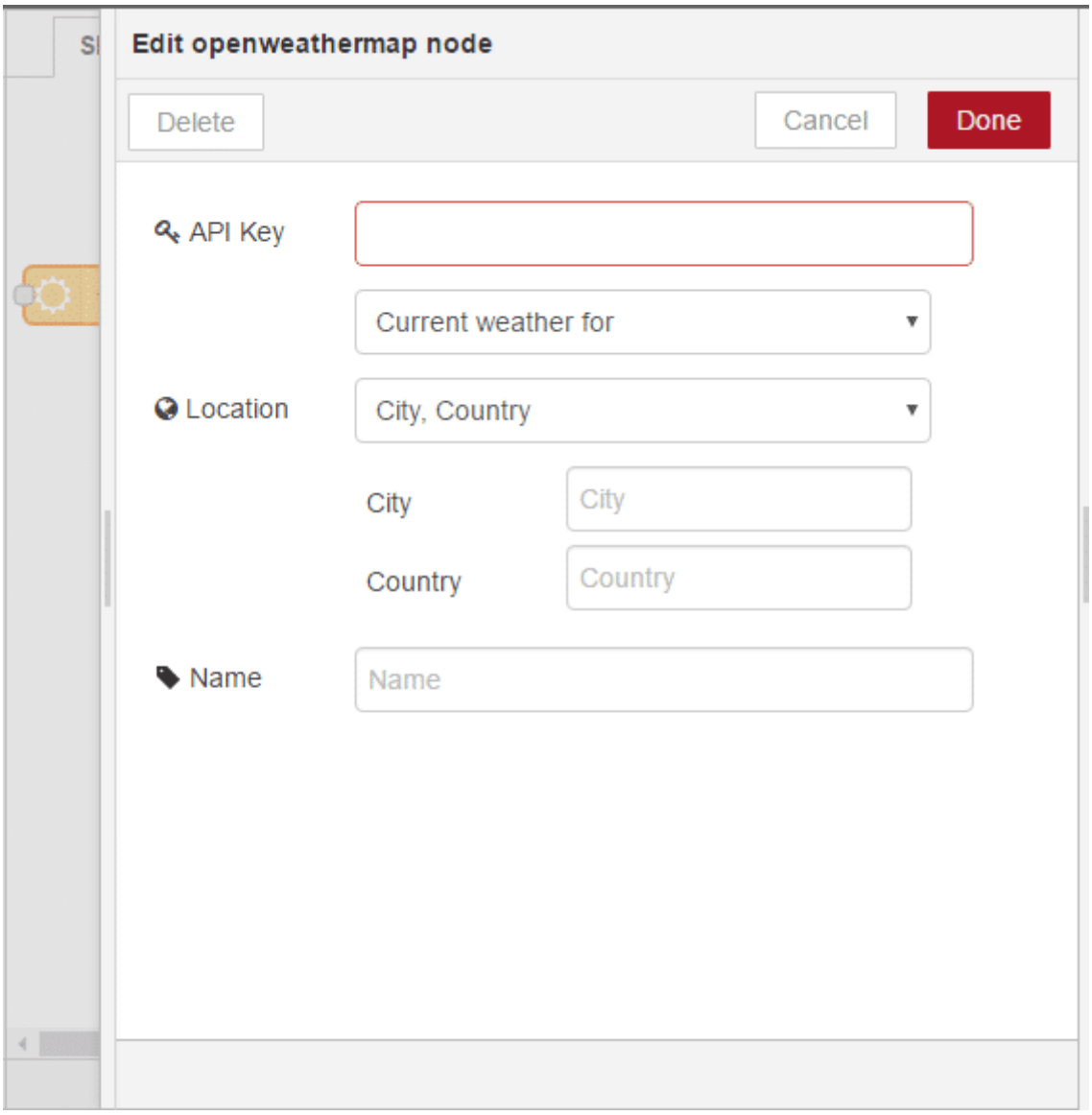
اذا قمت بتحديد لوحة info لظهور المعلومات على الجانب الايمن ، يمكنك أن ترى وصفا لعقدة openweathermap مع معلومات مفصلة حول كيفية إعدادها و استخدامها. ستلاحظ بعض الأمور المهم :

1– أن لديها بنية JSON كما ان لها msg.payload ، تحتوي على الكثير من التفاصيل حول الطقس على سبيل المثال : سرعة الرياح و

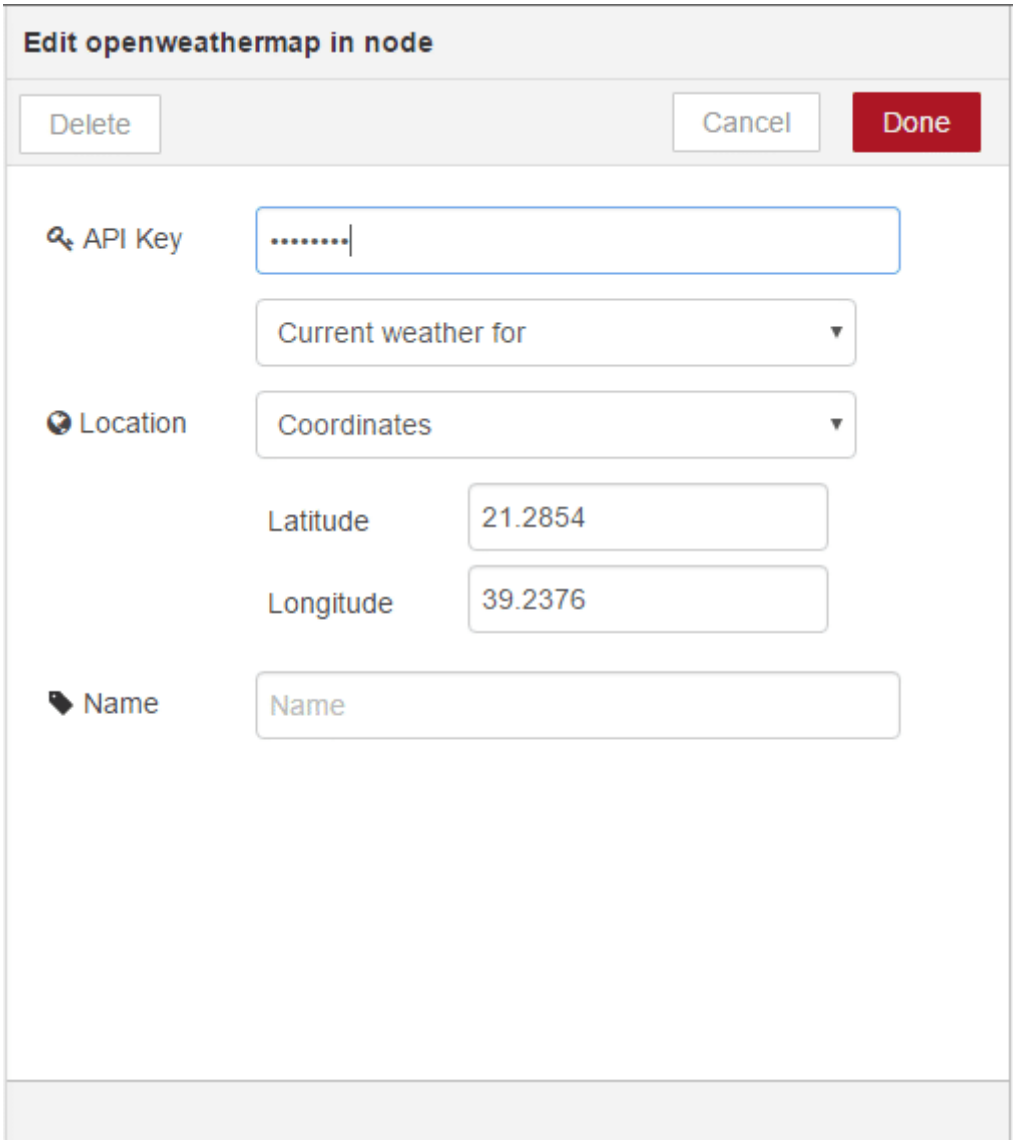
درجة الحرارة tempc.

2– العقدة تعرف ثلاث خصائص للرسالة الجديدة : msg.location ، msg.time و msg.data. وكما ذكر أعلاه، انت لك الحريه في إضافة خصائص الرسائل ، openweathermap اضافت هذة الخصائص الجديدة لتحمل معلومات إضافية مرتبطة مع الرسالة .

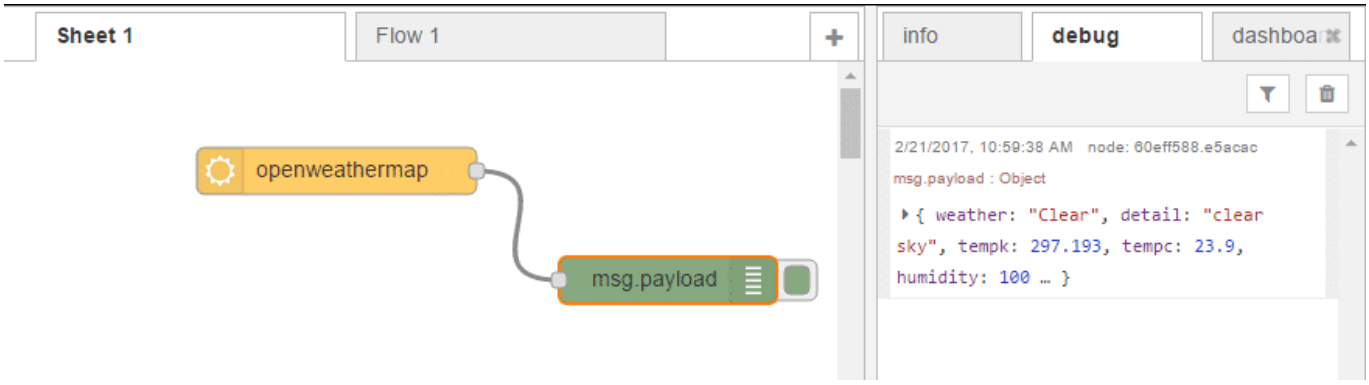
سنقوم بإعداد العقدة و النظر في بنية البيانات الفعلية التي يتم توليدها بعد أن تسأل عن الطقس المحلي الخاص بك . قم بالنقر المزدوج على العقدة وملء النموذج حسب موقعك. اكتب المدينة و الدولة في الحقول المخصصة او وضع احداثيات مدينتك. قم بإضافة API الذي تم الحصول عليه من http://openweathermap.org/appid ثم انقر موافق كما هو مبين بالشكل التالي:



في الصورة ادناه تم تحديد موقع مدينة جدة في المملكة العربية السعودية.

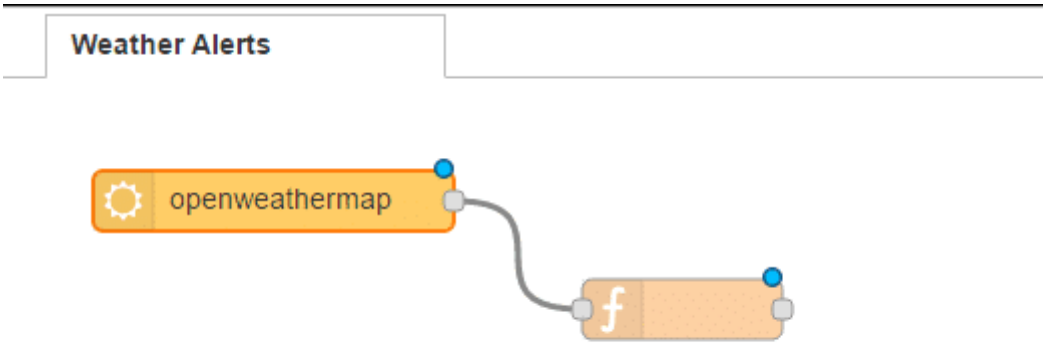


ثم قم بسحب و إسقاط عقدة debug و قم بتوصيل العقد معا عن طريق السلك قم بالضغط على زر Deploy لرؤية رسالة payload من عقدة openweathermap في لوحة debug .

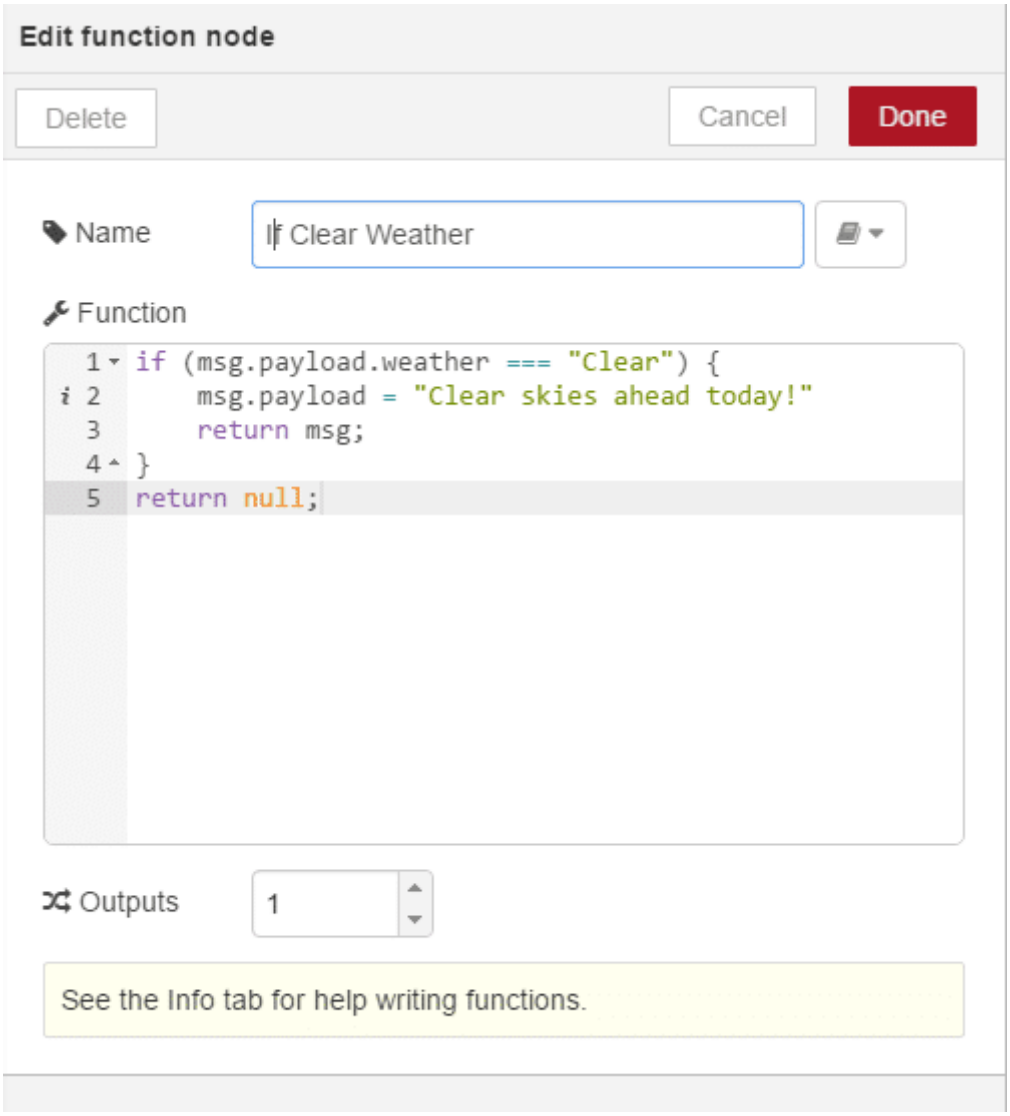


كما ترون في الصورة اعلاه على لوحة الاخراج، العقدة توفر الكثير من المعلومات حول الطقس المحلي الخاص بك، بالعادة تتوفر المعلومات كأزواج من اللإسم: القيمة . على سبيل المثال ، استخدام حقل الطقس "weather" لاختبار ما اذا كان الطقس جيد "Clear" . في حال كان جيد عليك ارسال تغريدة.

لبرمجة هذا المنطق، عليك استخدام عقدة function . تم عرضها في الدرس الأول ولكن دون ذكر التفاصيل. دعونا نرى كيف يتم استخدامها عن طريق سحب واحدة على مساحة عمل التدفق و النقر المزدوج لفتح محرر العقدة .



الان انقر نقرا مزدوجا فوق عقدة function وقم بكتابة الشرط التالي:

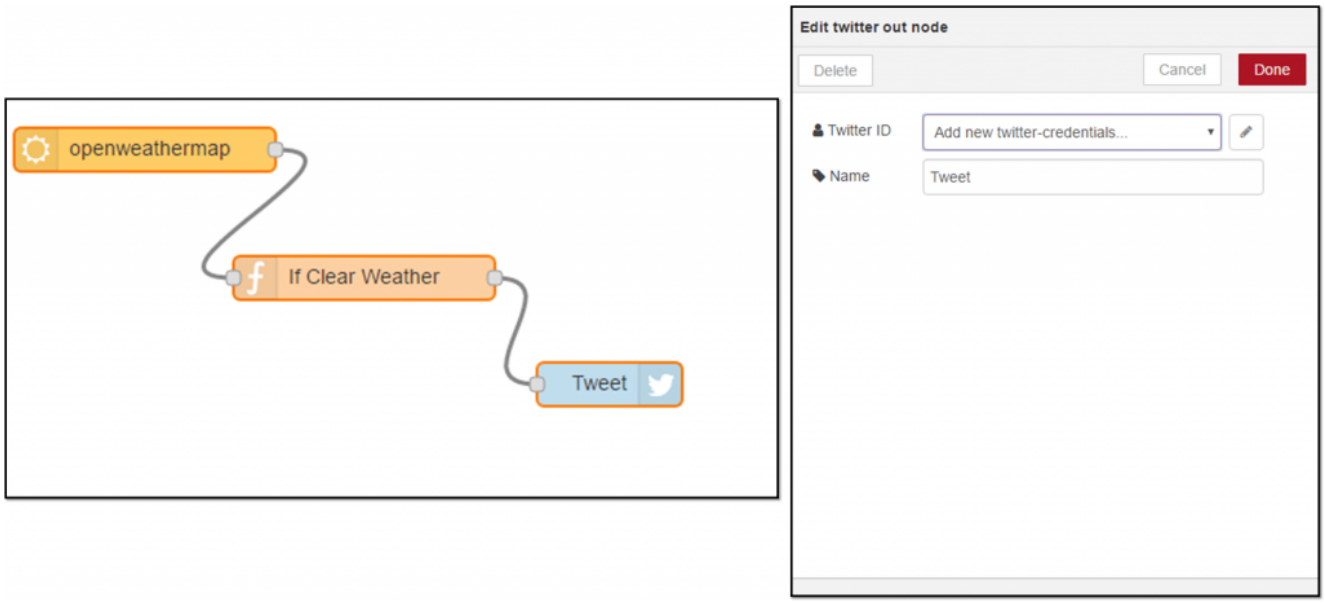


في الشرط تم التحقق ما اذا كان الطقس جيد "Clear" أو لا . اذا تحقق الشرط ، فإنه سيتم إعادة كتابة رسالة payload لتصبح "Clear skies ahead today!" (في السطر 2) . و اذا لم يتحقق سيعيد رسالة فارغة . السطر الأخير مهم لأن الـ Node-RED تتجاهل الرسائل الفارغة.

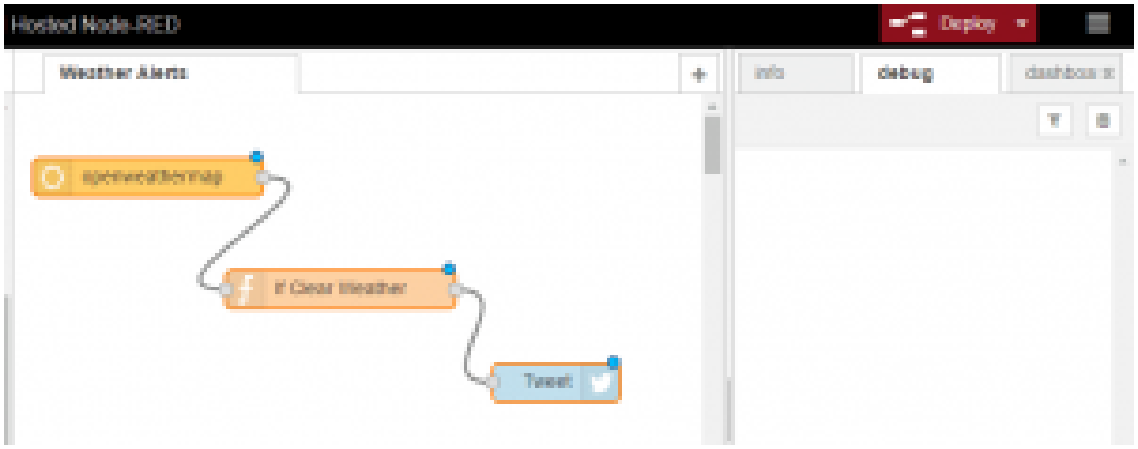
```
if (msg.payload.weather === "Clear") {
msg.payload = "Clear skies ahead today!"
return msg;
}
```

```
return null;
```

يمكنك الأن أن تفعل عدة تطبيقات على هذا التدفق مثلا توصيل هذة الرسالة إلى عقدة البريد الإلكتروني أو عقدة pushbullet . لهذا الدرس سيتم استخدام عقدة التويتر. قم بسحب عقدة التويتر على مساحة العمل ، انقر نقرا مزدوجا فوق العقدة و قم بملء اعتماد حساب التويتر الخاص بك كما هو موضح بالصورة ادناه :



وبمجرد الانتهاء من إعداد هذة الوثائق، تقوم Node-RED بتخزين هذة الوثائق التي يمكنك استخدامها في التدفقات المستقبلية. و أخيرا قم بالضغط على زر النشر Deploy و راقب حساب التويتر الخاص بك لرؤية التغريدة الجديدة في كل مرة يتم التحقق أن حالة الطقس جيدة .



ميزات الواجهة

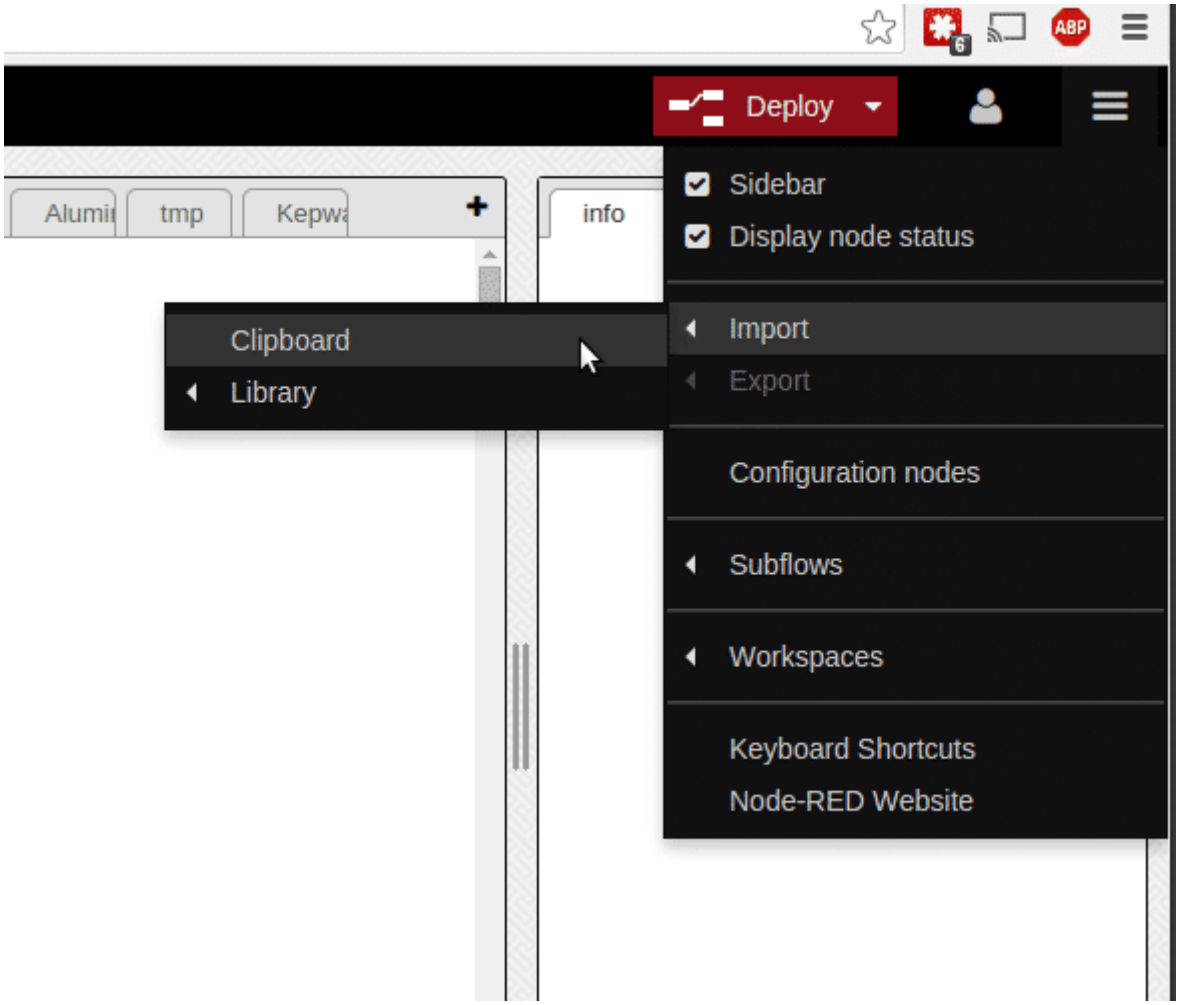
واحدة من أكثر السمات فائدة في واجهة المستخدم Node-RED هو القدرة على نسخ و لصق العقد و حتى التدفقات كاملة باستخدام الحافظة clipboard. تدفقات الـ Node-RED يتم تمثيلها بكود JSON و التي يمكنك تصديرها من مساحة العمل ، و إستيرادها إلى مساحة العمل بإستخدام القائمة المنسدلة في Node-RED على الجانب العلوي الأيمن من نافذة Node-RED .

بدلا من بناء التدفقات في هذا المثال، أو في الواقع في أي سلسلة من الدروس ، يمكنك ببساطة نسخها من موقعنا مباشرة و لصقها في مساحة العمل. بعد ذلك كل ما عليك هو تكوين الإعدادات بشكل صحيح.

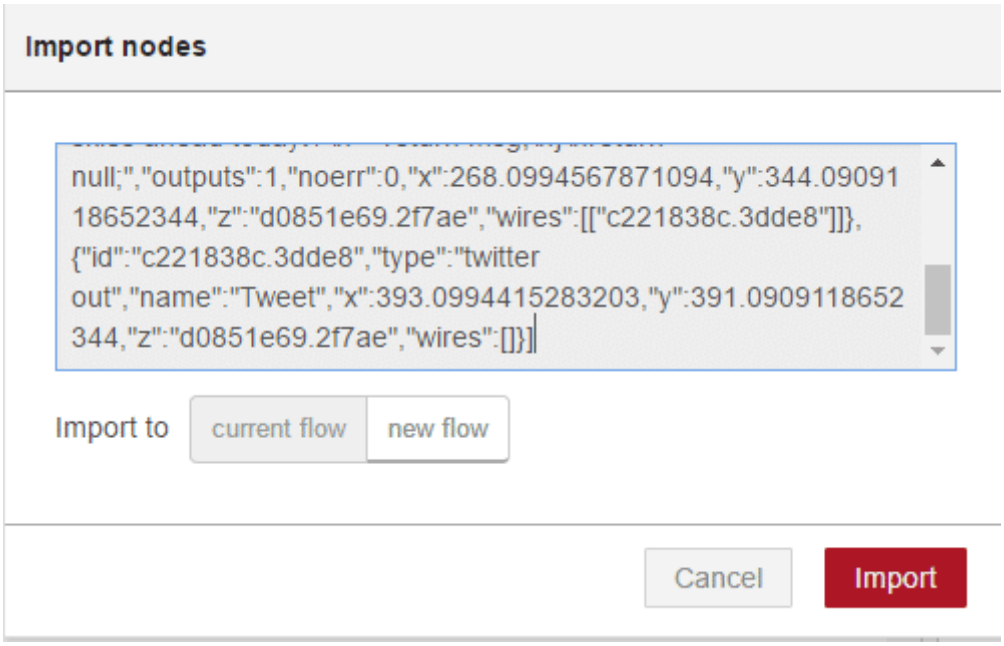
دعونا ننظر كيفية القيام بذلك.

قم بنسخ الكود التالي أدناه و عن طريق القائمة المنسدلة في أعلى اليمين في نافذة Node-RED اختر import ثم Clipboard كما هو موضح بالصورة أدناه :

```
[{"id":"3b4275ef.c4bd8a","type":"openweathermap
in","name":"","lon":"","lat":"","city":"","country":"","x":141.0994415283203,"y":284
.0909118652344,"z":"d0851e69.2f7ae","wires":[["b766183d.4899e8"]]},{"id":"b766183d.4
899e8","type":"function","name":"","func":"if (msg.payload.weather === \"Clear\")
{\n msg.payload = \"Clear skies ahead today!\"\n return msg;\n}\nreturn
null;","outputs":1,"noerr":0,"x":268.0994567871094,"y":344.0909118652344,"z":"d0851e
69.2f7ae","wires":[["c221838c.3dde8"]]},{"id":"c221838c.3dde8","type":"twitter
out","name":"Tweet","x":393.0994415283203,"y":391.0909118652344,"z":"d0851e69.2f7ae"
,"wires":[]}]
```



سوف يظهر لك صفحة حقل منبثق قم بلصق الكود في نافذة الإدخال ثم اضغط على موافق OK .



سيظهر تدفق جديد في مساحة عمل جديدة أو حالية حسب إختيارك. ويمكنك تهيئته كما لو أنك قمت بإنشائه بنفسك ثم قم بنشره .