بناء التدفقات عبر البروتوكولات

في هذا الدرس سوف نتطلع لرؤية المزيد من العقد الأكثر شيوعا و بناء مشاريع بإستخدام بعض العقد التي تم شرحها في الدروس السابقة . سوف نبدأ بسلسلة من الأمثلة القائمة حول بروتوكول MQTT و التي تبين كيفية ربط مجموعة من عقد معالج الرسالة . ثم سيتم النظر لطرق أخرى للحصول على الرسائل من و إلى التدفقات باستخدام بروتوكولات مثل TCP و UDP و مآخذ الويب websockets .

في نهاية هذة المحاضرة سيكون لديك فهم أفضل لبعض العقد الأساسية المستخدمة في تدفق Node-RED . وسيكون لديك القدرة على بناء التدفقات المعالجة المتطورة التي تأخذ الأحداث في العالم الحقيقي و معالجتها و توليد استجابات لإيصال النتائج خارج تدفق خاص بك .

السلسلة التالية من الأمثلة تبنى على عقدة mqtt ، الذي يوفر وسيلة ملائمة لإتخاذ المدخلات من وسيط MQTT . MQTT هو مثال على نظام نشر / اشتراك والذي يتيح لأجهزة الاستشعار بنشر التحديثات التي تسلمها إلى العميل المشترك في ذلك المستشعر . يستخدم MQTT نموذج الموضوع للسماح للناشرين (مثل اجهزة الاستشعار) لإنشاء الموضوعات و نشر البيانات للمواضيع ويمكن للأخرين الاشتراك في هذا الموضوع للحصل على اشعارات غير متزامنة من البيانات المرسلة لهذا الموضوع .

أنظمة pub/Sub هي طريقة رائعة لربط النظم الموزعة المتباعدة وهم خريطة جيدة لنموذج أنماط IOT حيث الأجهزة أو الأشياء تولد الأحداث التي ترغب في مشاركتها. بروتوكول MQTT ، فضلا عن كونها غير متزامنة هو أيضا خفيفة الوزن .

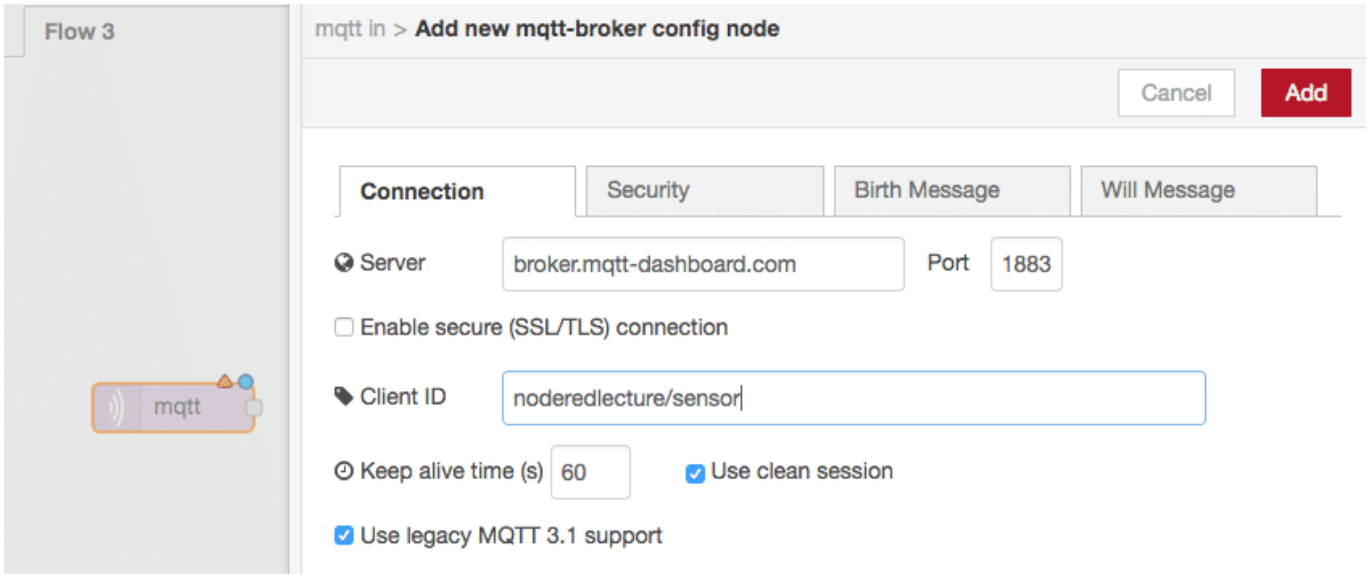
تلقي JSON عبر رسالة MQTT

لإستخدام عقدة MQTT ، تحتاج الوصول إلى الوسيط broker. هناك عدد من الخوادم تشغيلMQTT مجانية ، على سبيل المثال http://test.mosquitto.org/ ، او تلك التي سيتم استخدامها في هذا الدرس www.hivemq.com .

باستخدام عنوان الوسيط و الموضوع، يمكنك تكوين/إعداد عقدة المدخلات mqtt للإشتراك في هذا الموضوع . الأمر الذي ادى إلى إنشاء رسالة جديدة هو نشر البيانات الجديدة حول هذا الموضوع . و تحتوي الرسالة على معلومات عن البيانات المنشورة، وتتضمن البيانات نفسها في msg.payload و موضوع وسيط MQTT في msg.topic .

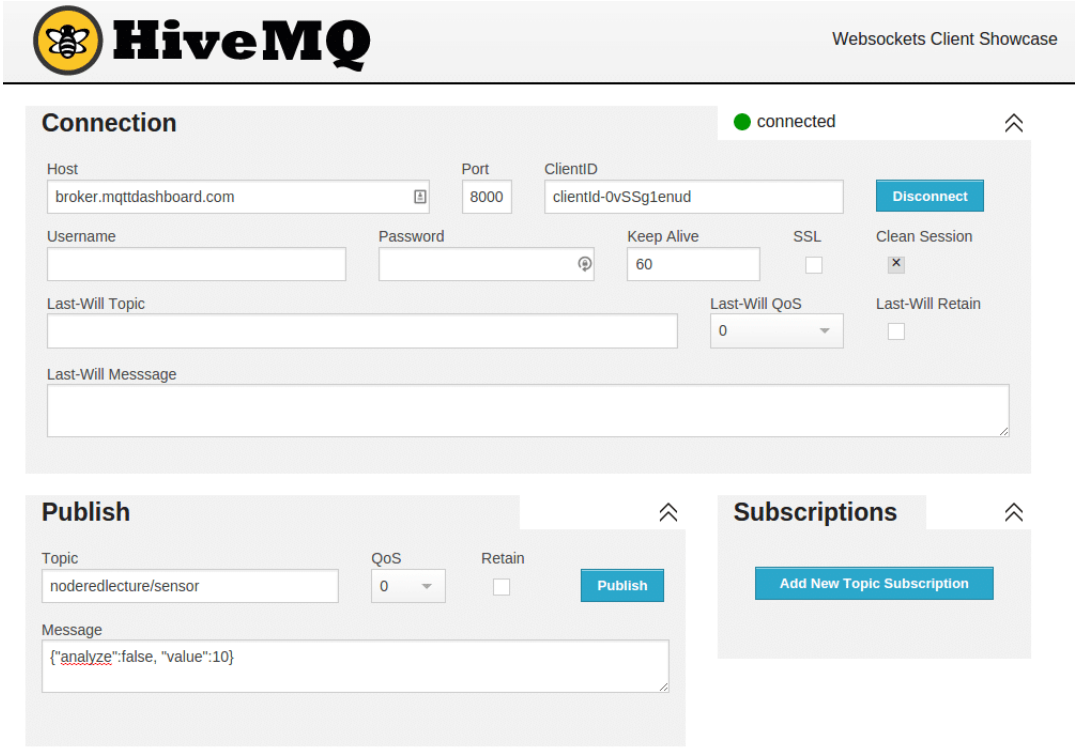
للبدء من عقدة mqtt ، عليك استخدام وسيط mqtt المجاني (hivemq) وهو متاح عبر
(http://www.hivemq.com/showcase/public-mqtt-broker/) . بالطبع يمكنك استخدام أي وسيط MQTT .

أولا، قم بسحب و إسقاط عقدة المدخلات mqtt إعدادها للوسيط. لا تنسى تكوين/إعداد الموضوع ليكون فريد من نوعه (unique)في هذا المثال نحن نستخدم noderedlecture/sensor ولكن عليك استخدام الموضوع الفريد الخاص بك ، مثال : <your name>/sensor .



هناك العديد من الطرق التي تمكنك من ارسال رسائل mqtt إلى hivemq . و يمكنك استخدام مآخذها الويب websockets للعميل (http://www.hivemq.com/demos/websocket-client/) ،
و لوحة الإعدادات لـ (http://www.mqtt-dashboard.com/dashboard) mqtt (mqtt dashboard) أو مكتبتك الخاصة .

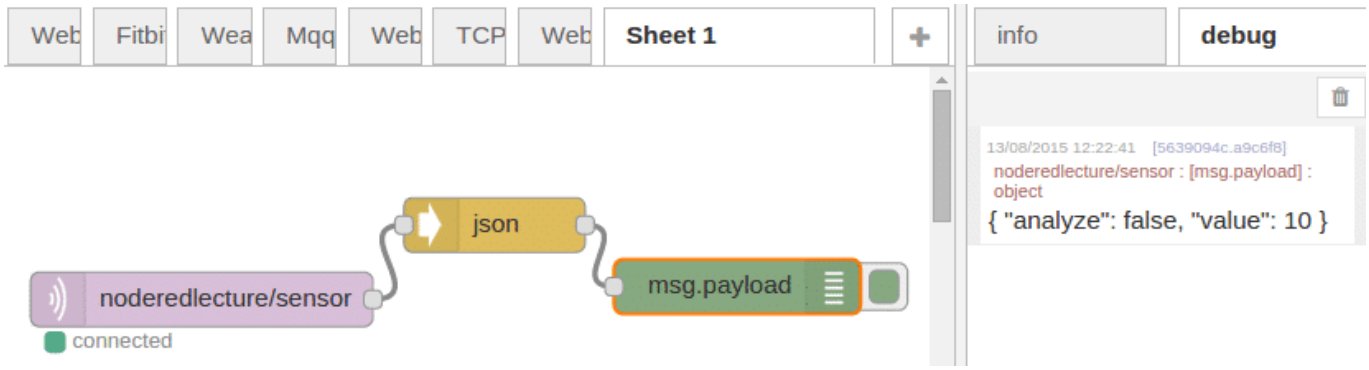
عليك استخدام مآخذها الويب للعميل websocket client في هذا المثال ، لذلك انتقل إلى تلك الصفحة و قم بالاتصال إلى الوسيط .
سوف تقوم بنشر سلسلة Json المرتبطة بالموضوع الذي قمت بتكوينه/إعداده ولرؤية ذلك قم بإستخدام عقدة qmtt وعقدة json .



بمجرد أن تقوم بإرسال سلسلة JSON ، سوف تحتاج إلى تحليل الرسالة التي تولدها عقدة mqtt عندما تتلقى رسالة MQTT . للقيام بذلك، سوف تحتاج إلى سحب و إسقاط عقدة json و توصيلها مع نقطة الإخراج في عقدة mqtt .

عقدة json هي نوع من convenience function ، حيث انه يوزع الرسالة الواردة و يحاول تحويلها من/إلى JSON .لذلك اذا قمت بإرسال سلسلة JSON سيقوم بتحويلها إلى جافاسكريبت JavaScript و العكس .

قم بتوصيل عقدة debug إلى عقدة json ثم قم بعملية النشر deploy ، استخدم لوحة إعداد HiveMQ لإرسال سلسلة JSON {{“analyze”:false, “value”:10} كما هو موضح بالشكل أدناه :

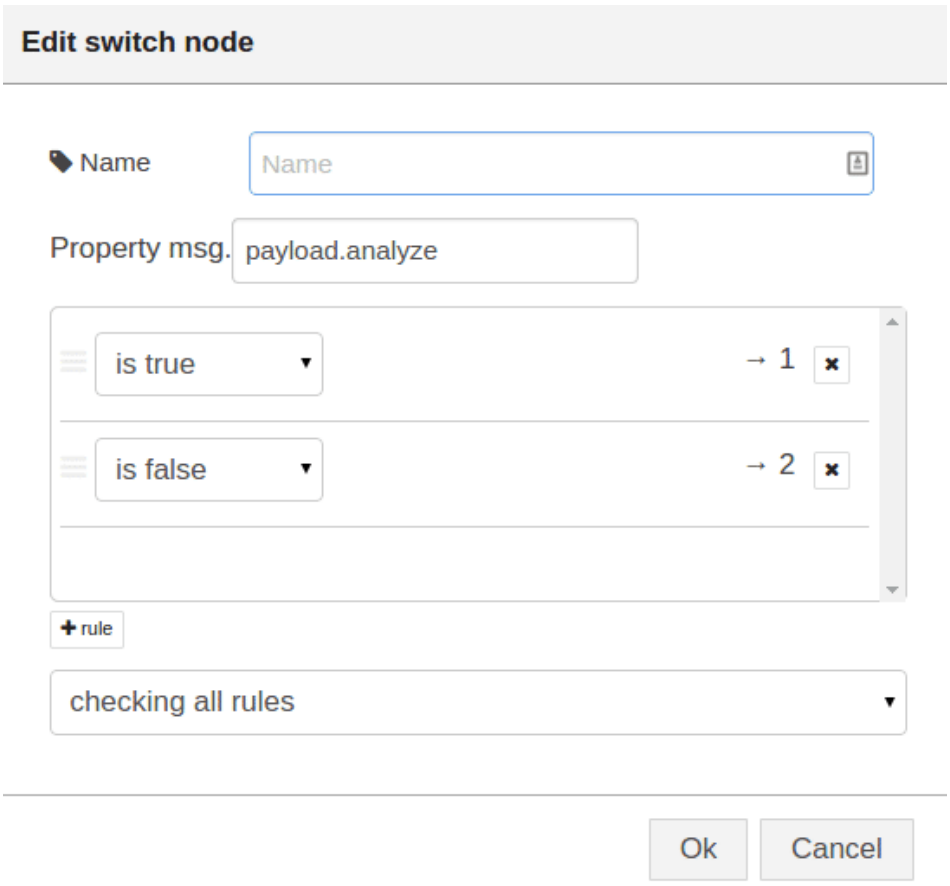


قم بالنظر عن كثب في المخرجات . يمكنك أن ترى msg.payload تحتوي على object و الذي يحتوي على حقلين حقل التحليل (analyze)و حقل القيمة (value) كما رأينا في الدرس السابق يمكنك الوصول إلى هذة الحقول عبر msg.payload.analyze و msg.payload.value . دعونا نلقي نظرة على العقدة وما يمكنها أن تفعل .

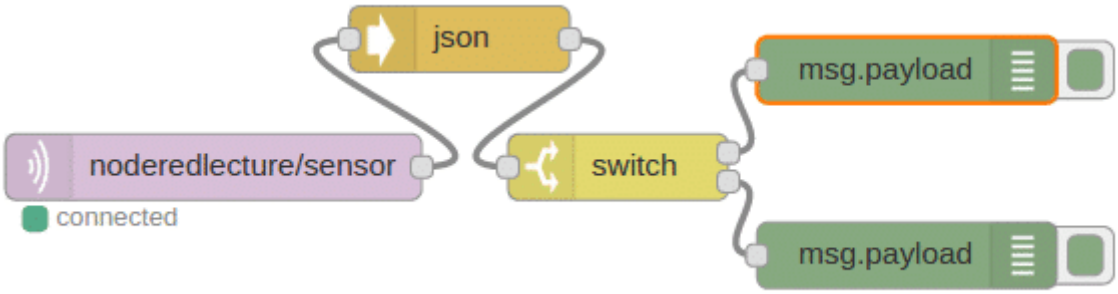
استخدام عقدة التبديل لتوجية الرسالة اعتمادا على خصائص معينة :

أحد المميزات الموجودة لدى JSON object أنها تتيح لك العمل بسهوله على خصائصها. سيتم استخدام عقدة التبديل (switch node) و التي تعمل كمفتاح أو توجيه للرسائل إلى اتجاه مخرج معين اعتمادا على خصائص الرسالة الواردة . على سبيل المثال، يمكنك تفحص خصائص الحقلmsg.payload.analyze و بالإعتماد على قيمتها (صحيحة / خاطئة) ، تقرر توجيه الرسالة إلى أحد مخرجات عقدة التبديل (switch Node) .

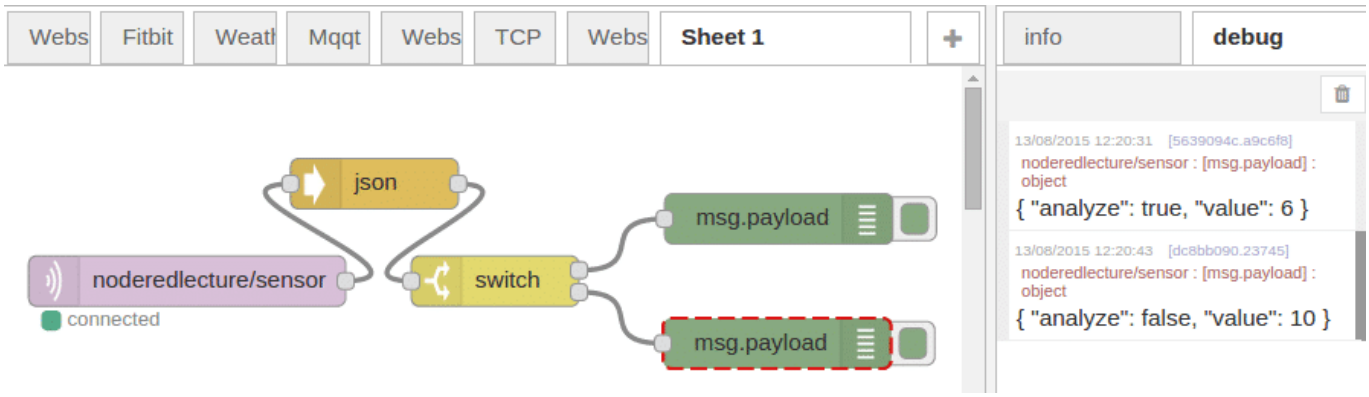
قم بسحب عقدة التبديل (switch node) ، ثم انقر نقرا مزدوجا عليها . قم بإعدادها للتحقق من قيمة الحقل msg.payload.analyze ، إذا كانت صحيحة سنقوم بإخراج الرسالة إلى المخرج الأول لعقدة التبديل ; و إذا كانت خاطئة سنقوم بإخراج الرسالة إلى المخرج الثاني لعقدة التبديل . قم بإعداد عقدة التبديل كما هو موضح بالصورة ادناه :



الآن يمكنك توصيل عقدتين إخراج debug كما هو موضح بالصورة التالية ، عند إعداد عدة مخارج للعقدة ، يتم ترقيم المخارج من أعلى ، لذلك المخرج 1 هو المخرج الأعلى و المخرج 2 هو المخرج السفلي في الشكل التالي.



قم بالرجوع إلى صفحة إدخال HiveMQ و إرسال رسالة {"analyze":true, "value":6} MQTT ، سوف تلاحظ أنه تم تفعيل المخرج الأول على عقدة التبديل و تم توجيه الرسالة الواردة على مخرج 1 ، إذا قمت بإرسال الرسالة التي تم استخدامها مسبقا {"analyze":false, "value":10} ، ستقوم عقدة التبديل (switch node) بتفعيل المخرج 2 و على عقدة الإخراج (debug) المربوطة على هذا المخرج ستظهر الرسالة. ستلاحظ على لوحة الاخراج عند تبويب debug على يمين النافذة طباعة الرسالة كما هو موضح بالصورة :

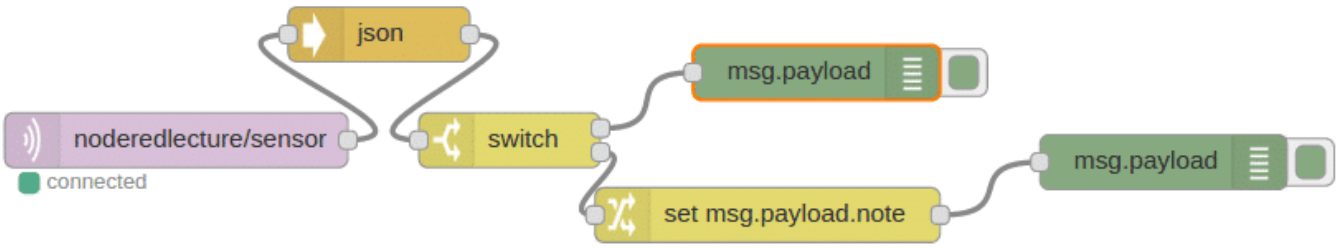


كما لاحظت، يوفر لك Node-RED عقد تسمح لك بسرعة تحديد محتويات الرسائل الواردة و توجيه الرسالة إلى أجزاء مختلفة من التدفق إعتمادا على محتوى معين للمدخلات.

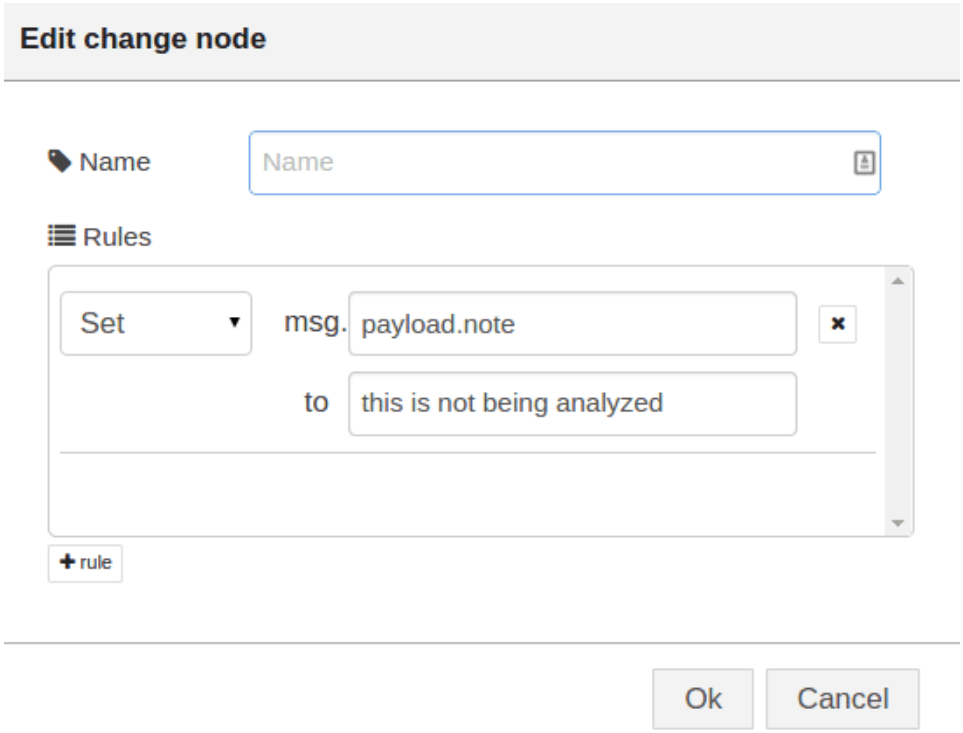
التلاعب في محتوى الرسائل بإستخدام عقدة التغيير :

عقدة أخرى مفيدة وهي عقدة التغيير (change node) ، و التي تسمح لك لتغيير محتوى الرسالة أو إضافة خصائص جديدة عليها . فيمكنك استخدام هذه العقدة للتأثير على الخصائص في الرسالة إما عن طريق تغيير خصائص موجودة ، أو حذف أو إضافة خصائص جديدة.

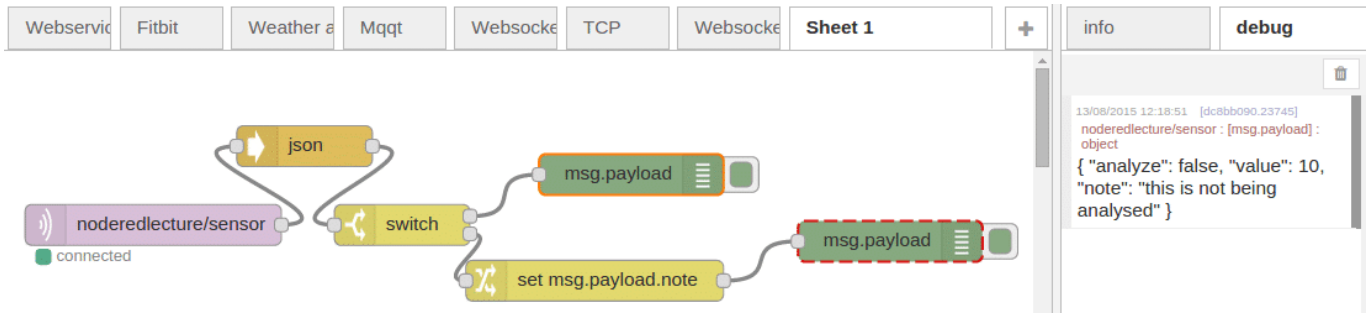
في هذا المثال ، سيتم إضافة حقل جديدة للرسالة وهو msg.payload.note . أولا، قم بسحب و إسقاط عقدة التغيير (change node) و توصيلها إلى المخرج الثاني من عقدة التبديل كما في الشكل أدناه ، كما تذكرون، ان محتوى حقل msg.payload.analyze الرسالة هو false فيتم اخراجها على المخرج رقم 2.



الان قم بإعداد العقدة لتعيين msg.payload.note لتحتوي على "this is not being analyzed" كما مبين لك بالصورة التالية :



عند استقبال الرسالة التي ترسلها عقدة التبديل (switch node) إلى المخرج 2 ، سوف يتم تعديلها لتحتوي على عنصر "note" مع جملة "this is not being analyzed" . اذا قمت بنشر (deploy) و اختبار التدفق بواسطة ارسال رسالة MQTT من HiveMQ ، سوف يظهر لك المخرج كما هو موضح بالصورة التالية:

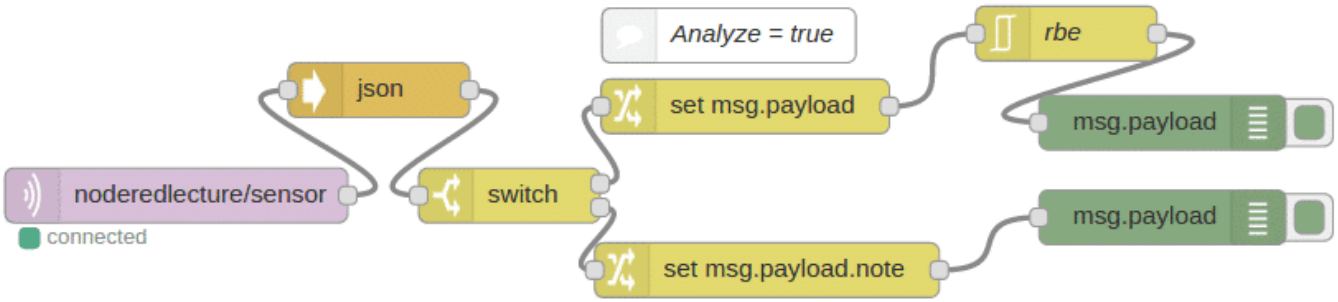


استخدام عقدة RBE :

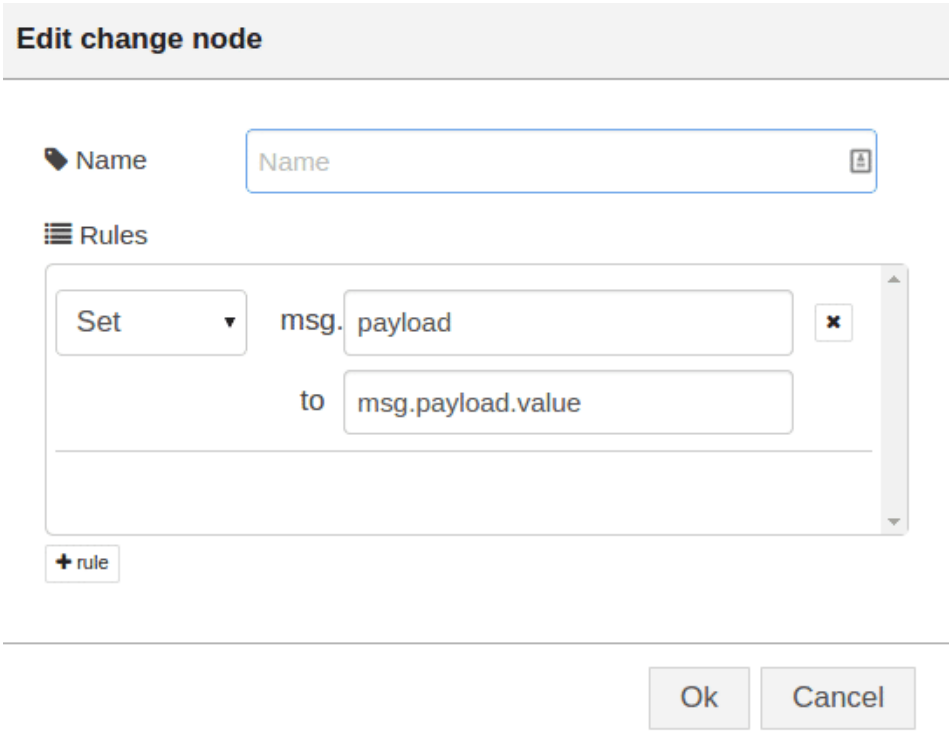
في هذا المثال، سيتم إضافة عقدة (report by exception -RBE) و التي تقوم بتمرير البيانات فقط اذا طرأ عليها تغيير . يمكنك تعيينها لفحص حمولة الرسالة (massage payload) و إما عن طريق تجاهلها حتى يطرأ أي تغيير على الرسالة (في حالة RBE) أو عندما تتغير الرسالة بنسبة معينة (في حالة deadband) .

في حالة استخدامRBE ، تعمل على الأرقام و الجمل . وفي حالة استخدام deadband فهي تعمل على أرقام فقط و يستخدم في إعداد الـ deadband نطاق + أو – ، بحث تكون القيمة الواردة يمكن أن تتقلب داخل نطاق معين قبل أن تنطلق .

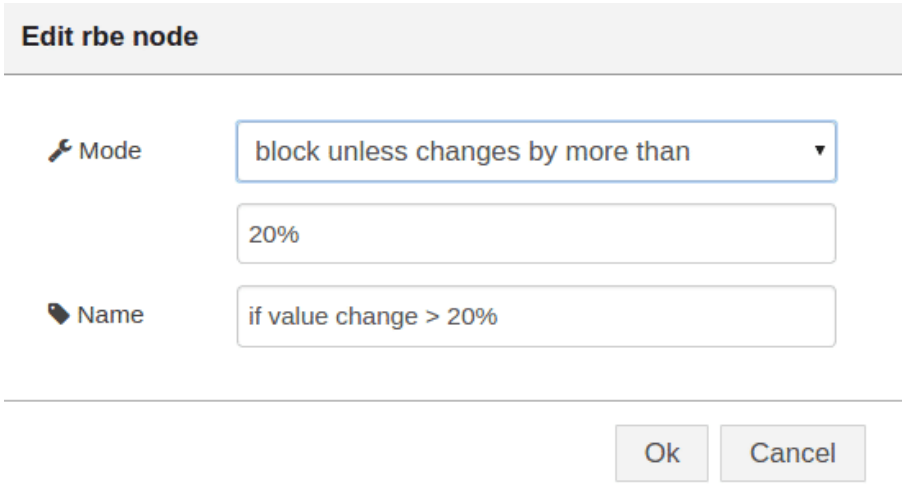
سنقوم بإضافة عقدة تغيير (change node) والتي سيتم توصيلها على مخرج 1 لعقدة التبديل (switch node) . ثم سنقوم بتوصيل عقدة rbe إلى عقدة التغيير كما هو موضح بالصورة أدناه . تذكر بأن الناتج يتوافق مع "analyze " ، وقد تم إضافة عقدة تعليق (comment node) و كتب عليها "analyze=true" ، لأنها مفيدة عند عمل تدفقات معقدة .



قم بفتح محرر عقدة التغيير (change node) لتعيين msg.payload إلى msg.payload.value . وهكذا يحدد المخرج من هذه العقدة إلى القيمة الموجودة في العنصر msg.payload.value من المدخلات الواردة كما في الشكل أدناه .

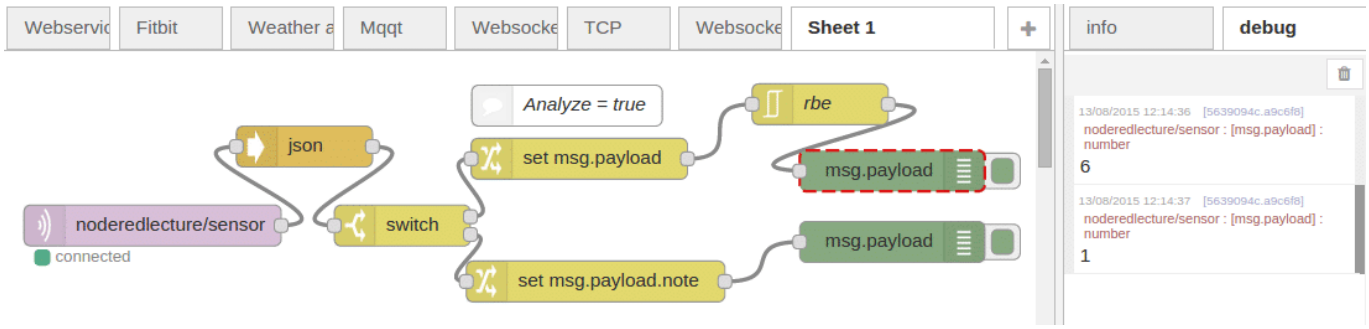


سنقوم بتوجية الرسالة إذا تم تغيير القيمة بنسبة 20% أو أكثر من ذلك ، فسوف تحتاج إلى النقر نقرا مزدوجا فوق عقدة RBE و لتجاهل الرسالة مالم تتغير القيمة بأكثر من 20 % .



لإختبار التدفق، قم بنشر deploy هذا التدفق ومن ثم العودة إلى صفحة HiveMQ و إرسال سلسلة من الرسائل. أولا ، سوف تحتاج إلى عيين قيمة analayze إلى true و بالتالي تقوم عقدة التبديل بإرسال الرسالة عبر المخرج الأول. إذا كنت تستخدم الرسالة التي تم تعينها

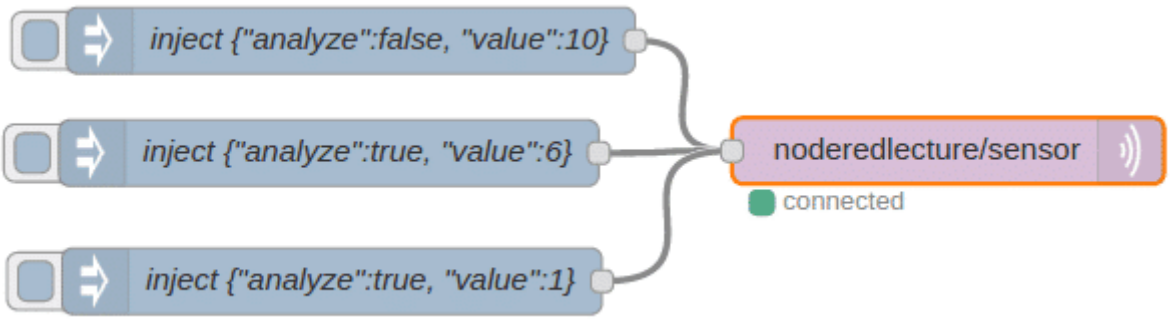
سابقة فإن القيمة (value) من 6 ، ستفشل في عبور عقدة RBE . ثم قم بإرسال رسالة ثانية من قيمة 10 ، فإن العقدة RBE تقييم الفرق ما بين 6 و 10 ، ويرها اكبر من 20 % ، فيقوم بإرسال رسالة إلى عقدة debug النهائية والتي ستطبع على لوحة الإخراج debug كما هو موضح بالصورة التالية :



استخدام عقدة الإخراج MQTT لإختبار التدفق :

كبديل لاستخدام صفحة الاختبار HiveMQ لنشر موضوع MQTT ، سنقوم بإعداد و تكوين عقدة اخراج mqtt . وهي تسمح لك لإعداد/تكوين خدمة MQTT و الموضوع الذي تريد نشره عليه. ويمكنك بعد ذلك إرسال عقدة الرسائل كسلسلة JSON كما كانت ترسل عبر صفحة الاختبار HiveMQ .

للعمل على ذلك، قم بسحب و إسقاط ثلاث عقد inject و عقدة اخراج MQTT كما هو موضح بالشكل التالي:



يمكنك الأن اختبار التدفق الذي قمت بإنشائه لتحليل رسالة MQTT مباشرة من مساحة العمل عن طريق النقر على زر الضح على الجانب الأيسر لعقد inject الثلاثة.

websockets

مآخذ الويب (Websockets) هي وسيلة أخرى مفيدة للقدرة على التواصل وهي مبينة ضمن عقد Node-RED عبر عقدة websocket . توفر مآخذ الويب (websocket) اتصال duplex TCP وقد صممت للسامح لمتصفحات الويب و الخوادم للحفاظ على "backchannel " والتي يمكن استخدامها لزيادة تفاعلات HTTP التقليدية ، مما يسمح للخوادم لتحديث صفحات الويب دون طلب العميل .

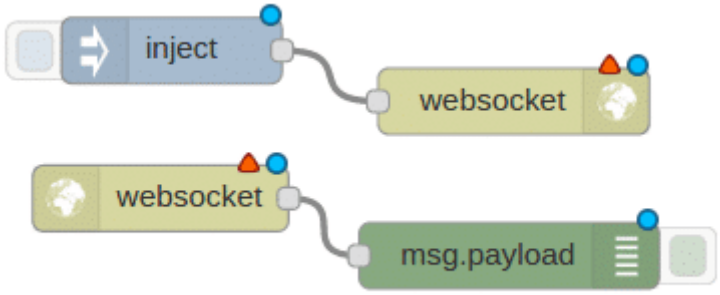
وعقدة websocket اما تكون للإدخال أو الإخراج ، مما يتيح لك الاستماع للبيانات الواردة (المدخلات) أو إرسال (الإخراج) على websocket . تم تصميم عقدة الإخراج للتحقق ولمعرفة ما اذا نشأت حمولة الانتاج على websocket في العقدة ، وفي هذة الحالة فإنه يستجيب إلى المرسل الأصلي . وإلا فإنه سوف يقوم ببث الحموله لجميع مآخذ الويب .

وبالإضافة إلى ذلك، يمكن إعداد كل من المدخلات و المخرجات لعقدة websocket إما أن تكون خادم أو عميل – في وضع الخادم

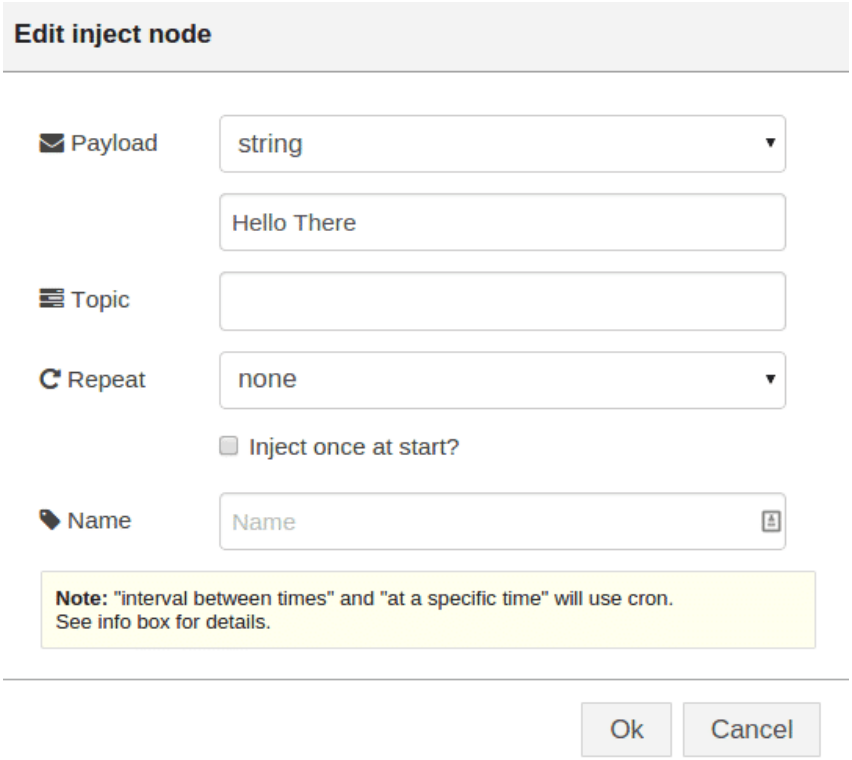
سيقوم بالإستماع إلى عنوان URL ، وفي وضع العميل سوف يتصل إلى عنوان IP معين .

لمعرفة كيفية عمل عقدة websocket ، عليك استخدام مآخذ الويب العامة التي تعمل على الموقع التالي: (https://www.websocket.org/echo.html).

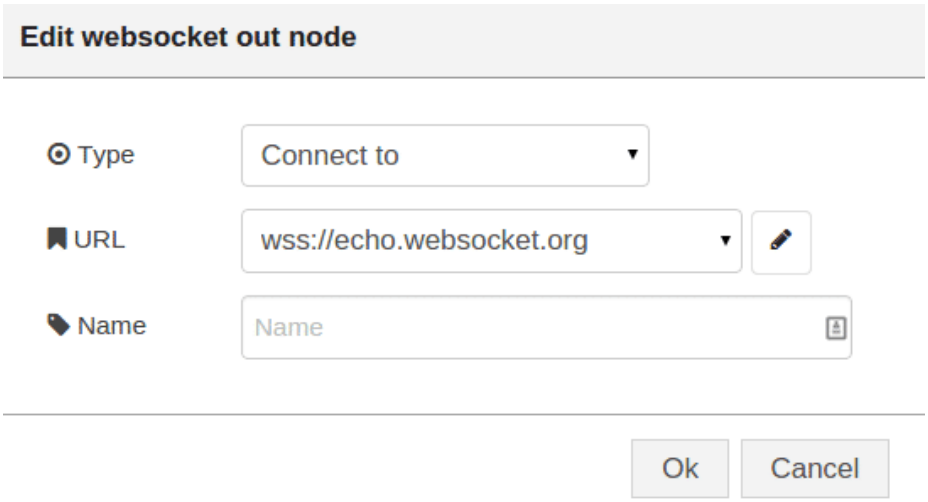
قم بسحب عقدة inject و عقدة إدخال و عقدة إخراج websocket و عقدة التصحيح debug ، ثم قم بتوصيل العقد كما هو موضح بالصورة التالية :



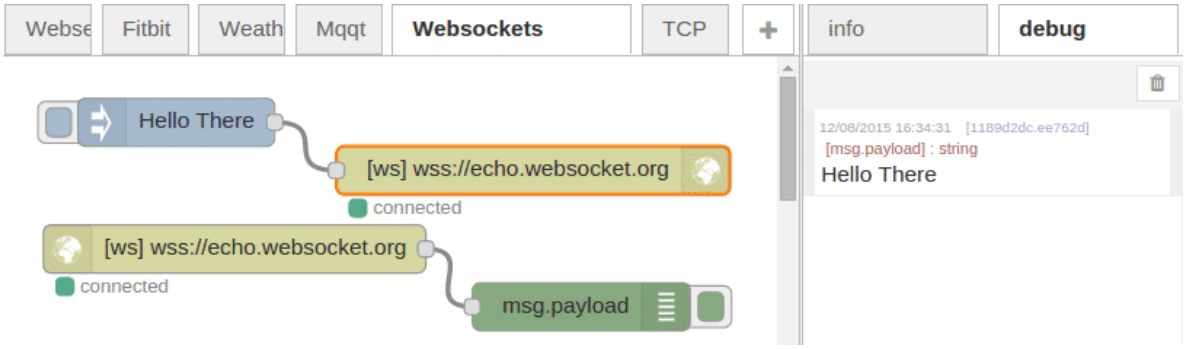
قم بإعداد عقدة inject لإرسال جملة payload تحتوي على "Hello There" كما في الصورة التالية :



قم بإعداد عقد websocket للإتصال إلى لإرسال wss://echo.websocket.org كما هو موضح بالصورة :



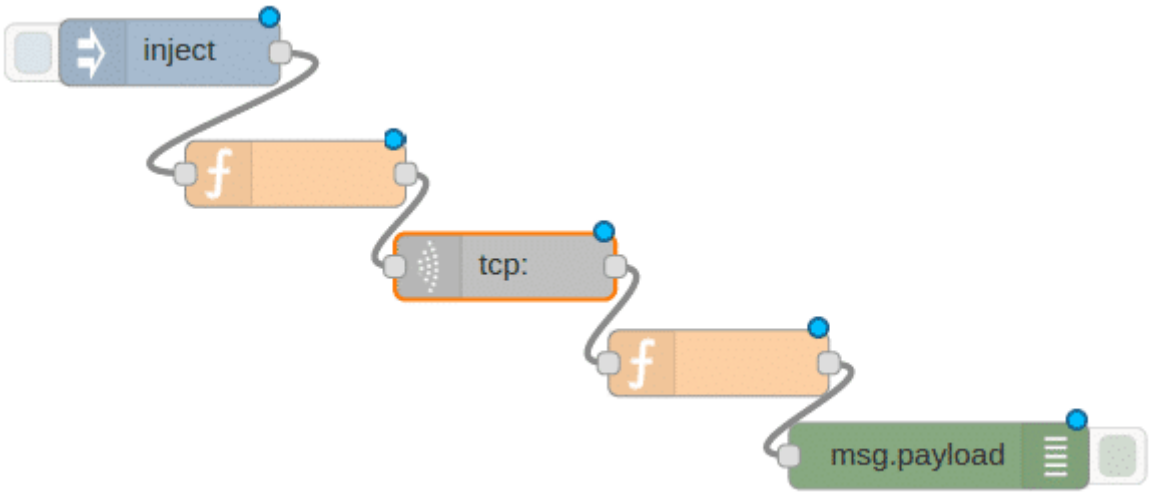
قم بنشر deploy التدفق ، عندما يتم الضغط على زر ضح عقدة inject سترى طباعتها كما هو موضح بالصورة :



إرسال طلبات TCP :

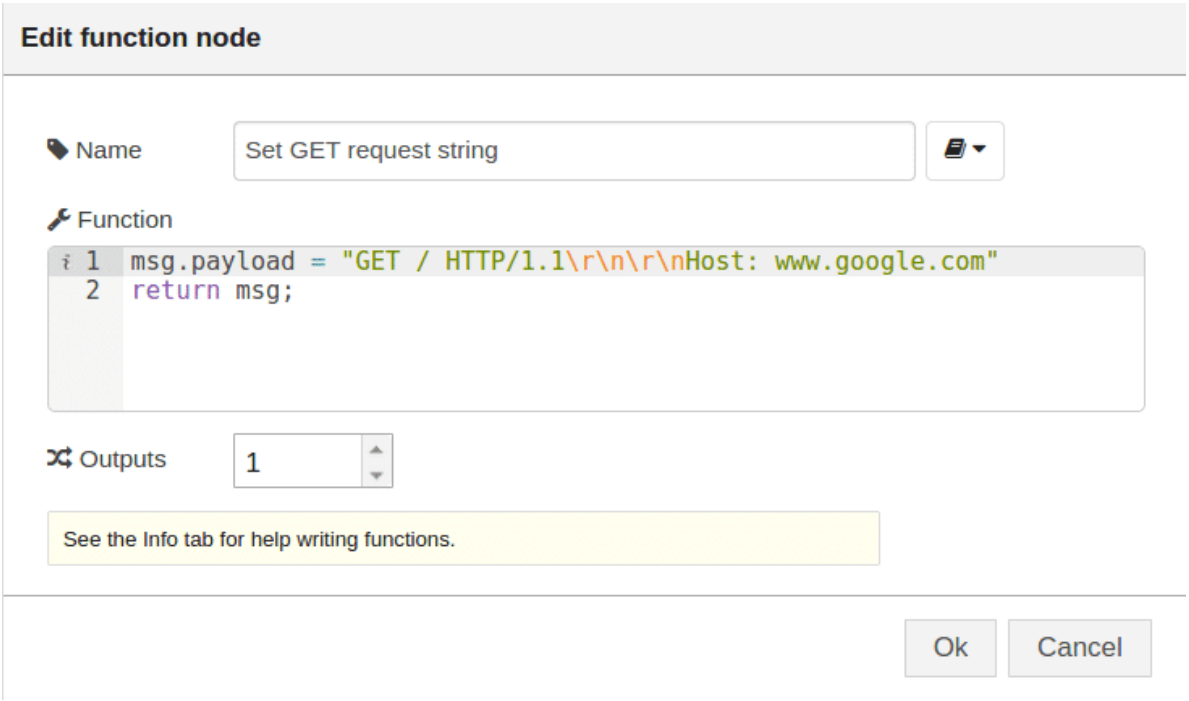
يوضح لك هذا المثال كيفية إرسال طلبات TCP باستخدام عقدة tcp . في هذه الحالة سوف تقوم بعمل طلب HTTP إتباعا للمواصفات في (http://tools.ietf.org/html/rfc2616#section-5.1.2).

يوضح هذا المثال استخدام عقدة tcp .حيث يمكن تكوين عقدة UDP أو HTTP لطريقة مماثلة . للبدء، دعونا نقوم بتوصيل عقد الـ inject و function و tcp و debug كما هو موضح بالصورة أدناه :

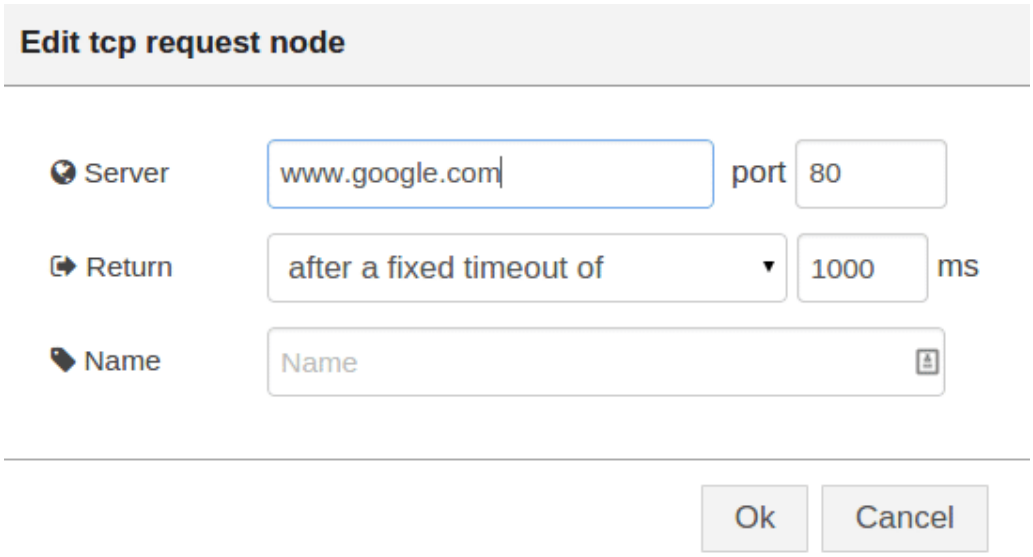


قم بتعديل على عقدة function للإضافة دالة و التي تضح جملة "GET / HTTP/1.1\r\n\r\nHost: www.google.com" كحمولة (payload) كما هو موضح بالصورة أدناه.

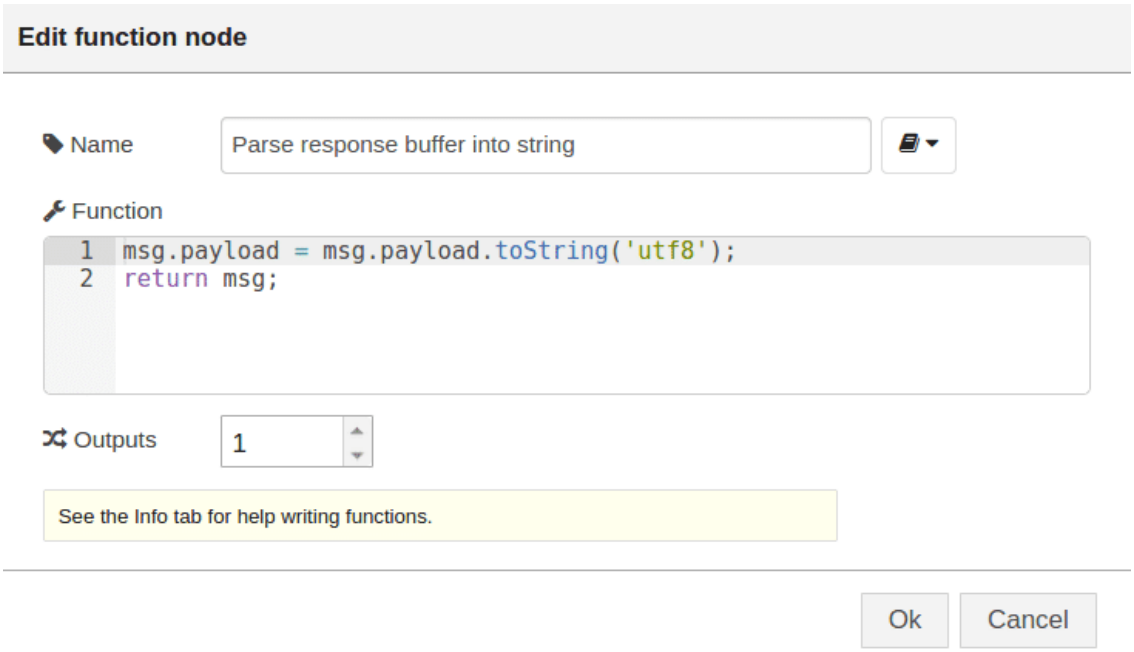
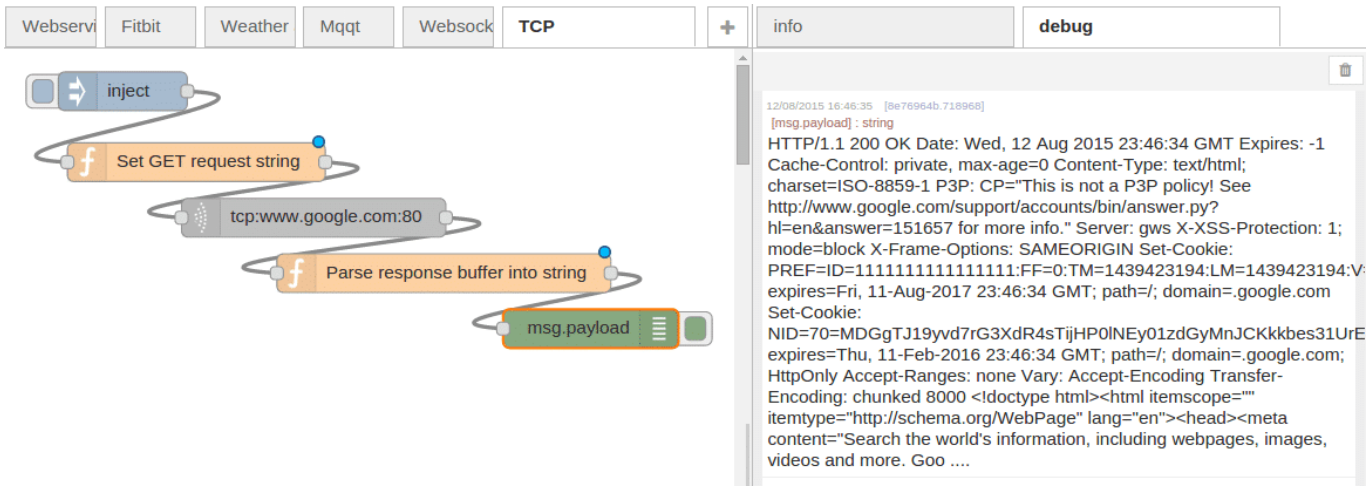
هذة سلسلة الجملة عبارة عن طلب HTTP ، مشيرا إلى انه طلب GET ، و البروتوكول هو HTTP 1.1 و المضيف (host) هو www.google.com . و \r\n\r\n\ هو زوج من إرجاع/ السطر الجديد (return/newline) و الذي يكون مطلوب في بروتوكول HTTP .



قم بإعداد عقدة طلب tcp للاتصال بالخادم www.google.com ، على المنفذ 80. و الإعداد لإغلاق الاتصال بعد 1 ثانية (1000 مللي ثانية) كما هو موضح بالصورة التالية :



إذا قمت بنشر deploy التدفق و الضغط على زر الضح في عقدة inject ، فسوف تقوم بعمل طلب إلى google و سوف تحصل على استجابة TCP . عقدة التصحيح debug ستقوم بطباعة الإستجابة كجملة كما هو موضح بالصورة التالية :



إذا قمت بنشر deploy التدفق و الضغط على زر الضح في عقدة inject ، فسوف تقوم بعمل طلب إلى google و سوف تحصل على استجابة TCP . عقدة التصحيح debug ستقوم بطباعة الإستجابة كجملة كما هو موضح بالصورة التالية :