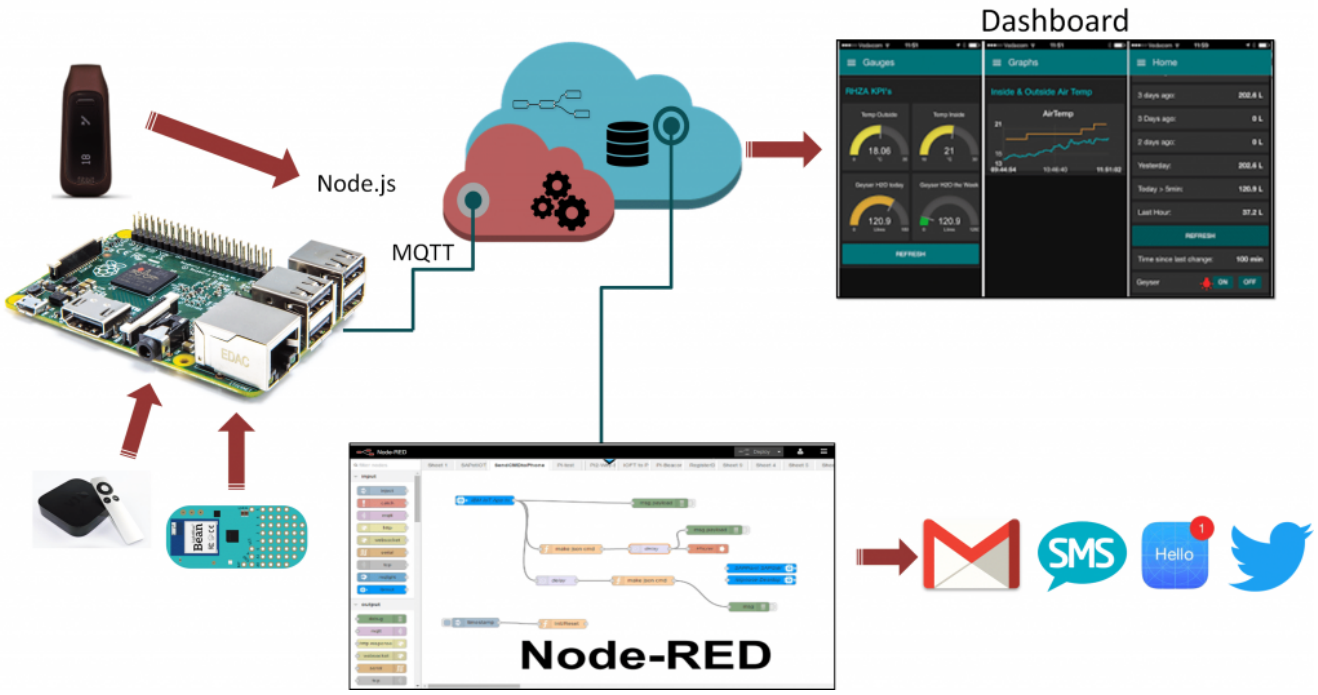
مقدمة موجزة للـ Node-RED

Node-RED

node-RED هي أدارة قوية لبناء تطبيقات انترنت الأشياء مع التركيز على تبسيط توصيل الأسلاك معا بين blocks التعليمات البرمجية لتنفيذ المهام. حيث يستخدم نهج البرمجية المرئية التي تسمح للمطورين لربط blocks التعليمات البرمجية المحددة مسبقا، و المعروفة بإسم العقد "nodes" ، معا لأداء المهمة. العقد "Nodes" المتصلة ، عادة ما تكون مزيجا من عقد الادخال(input nodes) ، عقد المعالجة (processing nodes) ، و عقد الاخراج(output nodes) ، يتم توصيلهم معا لتشكيل التدفقات "Flows" .

في الأصل تم تطويرها كمشروع مفتوح المصدر من قبل IBM، لتلبية حاجتهم للاتصال السريع بين الأجهزة و خادمات الويب web services)) و البرامج الأخرى، وسرعان ما تطورت لتكون الغرض العام لأداة برمجة IOT . وعلى الرغم من أن Node-REDمصممة أصلا للعمل مع انترنت الأشياء، أي الأجهزة التي تتفاعل و تتحكم بمحيطها، كما تطورت و أصبحت مفيدة لمجموعة واسعة من التطبيقات.



في هذا الدرس سيتم عرض بعض الأمثلة للـ Node-RED ، لإدراك قدراتها و تزويدك بالمفاهيم الأساسية. سنقوم أيضا بمناقشة تاريخ Node-RED و أصولها لإعطائك فكرة عن ما هو Node-RED و ماهي أنواع المهام التي يمكنك تنفيذها من خلاله.

بنهاية هذا الدرس يجب أن يكون لديك فهم على مستوى عال للـ Node-RED ، و نموذج البرمجة، و بعض الدمج البسيط للعقد.

دعونا نبدأ من خلال النظر إلى بعض الأمثلة البسيطة لأنواع الأشياء التي يمكنك القيام بها باستخدام الـ Node-RED . الهدف من هذا

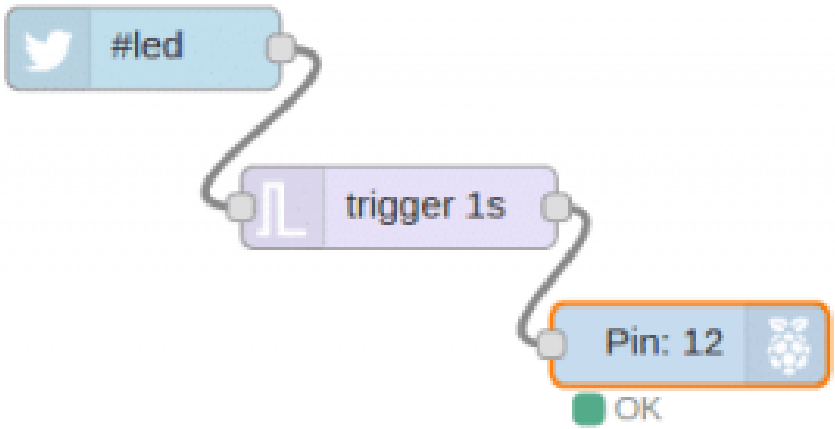
الدرس ليس تحصيل البرمجة باستخدام Node-RED ، ولكن لتتمكن من معرفة قدراتها التي يمكن من خلالها تنفيذ المهام من خلال تقديم بعض الأمثلة على مستوى عال. اذا كنت تريد البدأ مباشرة ، يمكنك القفز إلى الدرس الثاني هناك سيتم تقديم خطوات لإنشاء و تشغيل التدفق الأول .

في أول مثال للتدفق في هذا الدرس ، سيتم استقبال رسالة من التويتر تحتوي على علامة hashtag للتحكم في تشغيل و ايقاف الـ LED عند وصول تغريبدة جديدة. على سبيل المثال دعونا نفترض أن Node-RED يعمل على Raspberry pi .

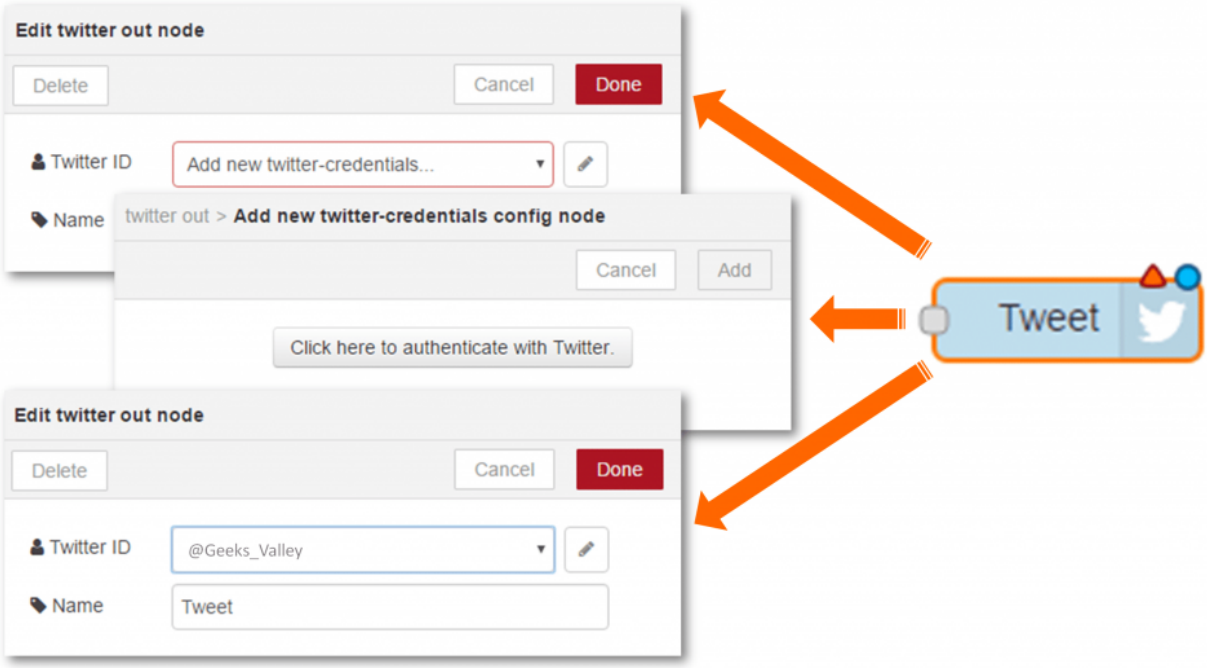
المثال الأول: استخدام التويتر للتحكم بالراسبيري باي

يتم انشاء تدفق الـ Node-RED للتحقق من التويتر و لمعرفة ما اذا تم الكشف عن hashtag #led، و التحكم بالـ LED الموصل على الراسبيري باي كما هو موضح بالشكل أدناه . كما تلاحظ ، يتكون التدفق من ثلاث عقد متصلة معا – عقدة التويتر على اليسار ، و عقدة trigger في الوسط ، و عقدة الراسبيري باي (gpionode) على اليمين. كل العقد ممثلة بمستطيلات وعادة ما تحتوي على رمز لتمثيل نوعها ، على سبيل المثال عقدة التويتر تحتوي على رمز التويتر (الطائر) و مع اسم النص الذي يمكن تعيينها من قبل المبرمج.

كل من هذة العقد بنيت في أداة البرمجية المرئية Node-RED ، و يمكن سحبها من لوحة العقد إلى مساحة العمل الرئيسية .ثم يتم توصيلهم معا عن طريق دمج /ربط علامة التبويب الخاصة بالإخراج إلى علامة التويب الخاصة بالإدخال من العقدة التالية في التدفق(النقط الرمادية الصغيرة على يمين و يسار مربعات العقد). في المحاضرة القادة سيتم عرض واجهة المستخدم بالتفصيل .



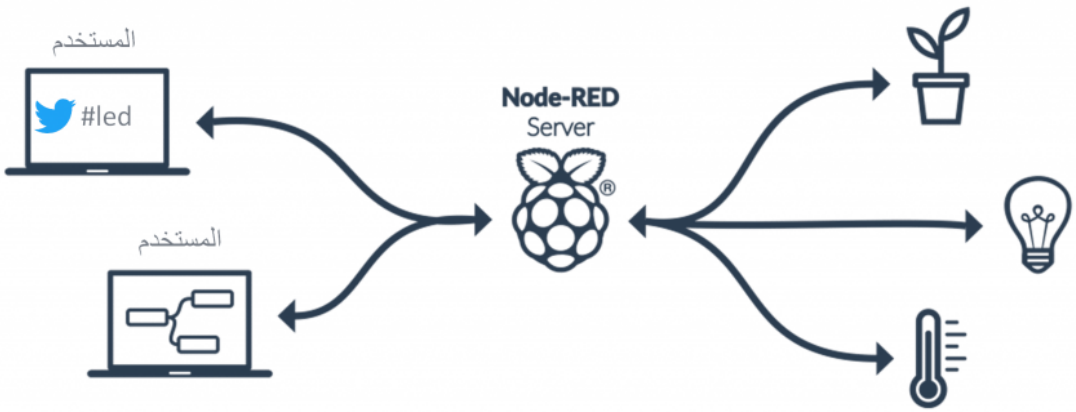
عقدة التويتر هي عقدة مضمنة في Node-RED و تخفي كل من تعقيدات استخدام تويتر API . يمكن تكوينها / تهيئتها باستخدام بيانات اعتماد حساب المستخدم و مجموعة متنوعة من سلاسل البحث، في حالتنا تبحث عن العلامة hashtag “#led”. عندما تكتشف عقدة التويتر هذة العلامة في حساب تويتر المستخدم، فإنها تنشئ رسالة جديدة تحتوي على تفاصيل التغريدة، و التي يتم تدفقها إلى العقدة التالية في التدفق. في مثالنا ، عقدة trigger تستقبل الرسالة مع تفاصيل التغريدة.



عقدة Trigger هي ايضا عقدة مضمنة في Node-RED و السلوك الافتراضي الخاص بها هو انتظار اي رسالة على مدخلاتها. عندما تتلقى الرسالة، فإنها تطلق اشارة ، و ترسل رسالة على تبويب الاخراج لديها مع قيمة "1" كنص للرسالة. ومن ثم ينتظر لمدة ثانية و يرسل رسالة ثانية مع قيمة 0 في نص الرسالة.

و نظرا لربط عقدة الـ trigger إلى عقدة الـ gpionode، التي تسيطر على المداخل و المخارج على الراسبيري باي ، و التي تتلقى هذه الرسائل الاثنين على مدخلاتها.

مرة أخرى، gpionode هي عقدة مضمنة في Node-RED و التي تخفي تفاصيل اعدادات منافذ الادخال و الإخراج على الراسبيري باي. فعندما تستقبل رسالة مع قيمة "1" في نص الرسالة ، فإنها تعطي المنفذ قيمة high، و عندما تتلقى رسالة مع نص بقيمة "0" فإنها تعطي قيمة LOW للمنفذ. في هذا المثال ، يتم تكوين عقدة gpionode للتحكم على نفذ رقم 12 .

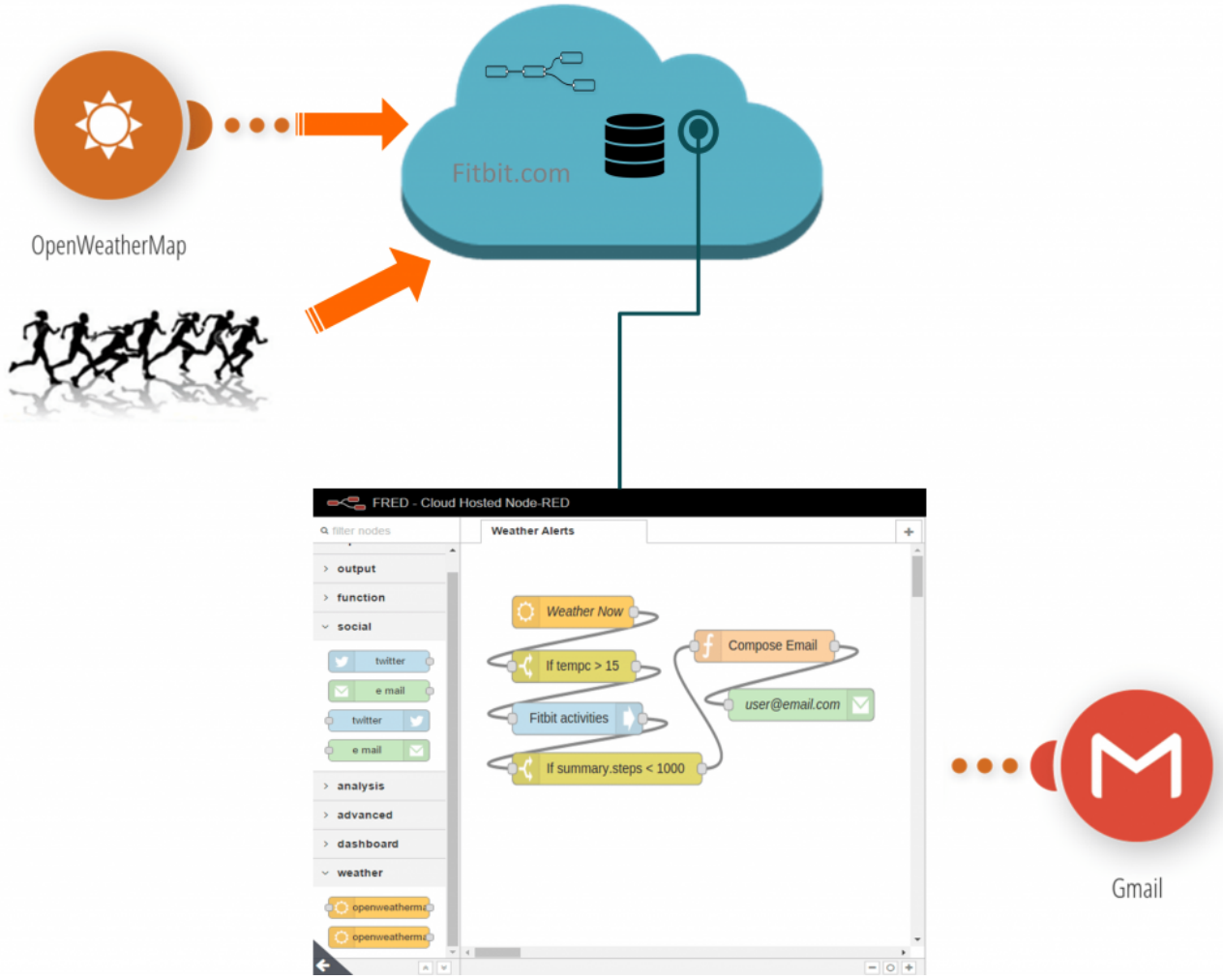


اذا كان لديك راسبيري باي مع LED موصل على المنفذ رقم 12 ، عقدة gpionode ستقوم بإخراج قيمة HIGH لمدة ثانية واحدة ثم قيمة LOW ، و سوف يسبب وميض على الـ LED لمدة ثانية واحدة.

اذا كنت تفكر في ما يحدث هنا، في الوقع هذا مثال قوي من Node-RED و انترنت الاشياء. فهو يكون ربط "Twitterverse" الى الراسبيري باي و LED ، وهو لا يتطلب اي برمجة- يتم تنفيذ جميع المهام من قبل العقد التي تم انشاءها في Node-RED و التي تحتاج فقط للتهيئة/الاعدادات.

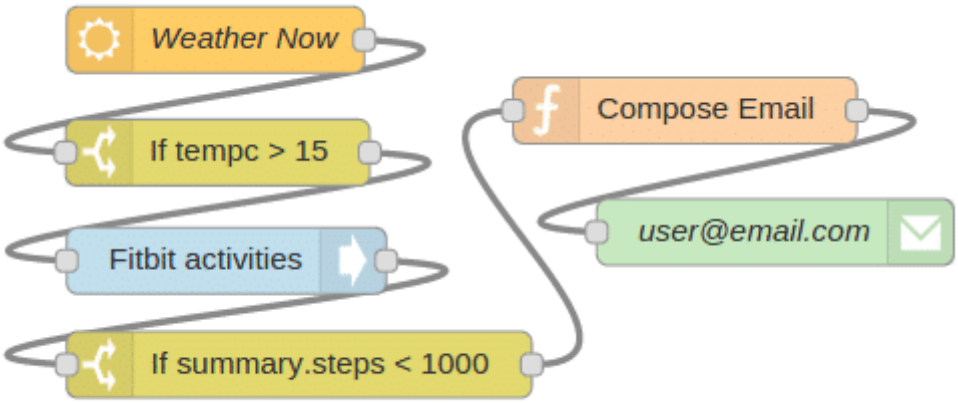
استخدام Node-RED لتنبيهك عند عدم انتهائك من اداء برنامج التمارين الرياضية

يظهر مثالنا الثاني على استخدام مختلف للـ Node-RED . بدلا من السيطرة على جهاز مثل الراسبيري باي ، هذا التدفق يجمع معلومات الطقس و استخدم fitbit الخاص بك لإرسال رسالة تنبيه لك عبر البريد الإلكتروني عندما يكون الطقس جيد و كنت قد تخلفت عن أداء برنامج التمارين الرياضية الخاصة بك.



التدفق هنا اكثر تعقيدا، وذلك بإستخدام 6 عقد (انظر الشكل أدناه) ، فعن طريق استخدام العقد المدمجة يمكنك إنجاز المهام المعقد مع برمجة قليلة أو معدومة.

كما ذكرنا سابقا ، يتم سحب العقد من لوحة العقد في واجهة المستخدم Node-RED و ربط العقد معا على لوحة التدفق (مساحة العمل الرئيسية) . بدءا من أقصى اليسار، دعونا نبدأ من عقدة openweather و التي يتم إنشاءها للتحقق من حالة الطقس في أي مدينة من المدن الكبرى في فترات محددة. عندما يتم التحقق من الطقس، تقوم بحزم النتائج في الرسالة التي يتم توجيهها إلى العقدة التالية، في هذه الحالة، إلى عقدة التبديل (switch Node) .



عقدة التبديل (Switch Node) تنفذ اختبار بسيط، فهي تقوم بالتحقق من قيمة الرسالة المدخلة اليها ثم تقوم بإرسال مع قيمة "1" اذا كان الاختبار صحيح .في حالتنا تم تكوينه لاختبار قيمة درجة الحرارة (tempc) لمعرفة ما اذا كانت 15 درجة مئوية أو أعلى.

عندما تتحقق عقدة التبديل (Switch Node) ويكون الاختبار صحيح، فإنها ترسل رسالة إلى عقدة fitbit و التي تمثل العقدة الثالثة في التدفق. عقدة fitbit يمكنك تكوينها / تهيئتها للحصول على إحصاءات على الجهاز الخاص بك من البوابة fitbit.com . تقوم عقدة fitbit بحزم الإحصاءات لجهازك في رسالة و توجيهها إلى العقدة التالية في التدفق .

العقدة التالية هي عقدة تبديل (switch Node) ايضا . تم تكوينها لمعرفة ما اذا كان عدد الخطوات التي قمت بمشيها اليوم هو أقل من 1000 . وذلك من خلال اختبار لمعرفة ما اذا كان حق البيانات summary.steps ، fitbit ، هو أقل من 1000. تذكر ، انه تم الحصول على هذه الإحصاءات في رسالة من عقدة fitbit و التي بدورها حصلت عليها عن طريق إجراء استدعاء للـ fitbit API على fibit.com .

اذا كنت قد مشيت اقل من 1000 خطوات في اليوم، فإن عقدة التبديل (switch Node) تولد رسالة التي يتم تمريرها خلال التدفق إلى عقدة function .

عقدة function هي عقدة عامة لجميع الأغراض التي تسمح لك لكتابة منطق البرمجة الخاص بك في جافا سكريبت (Javascript) . وسوف تناقش هذه العقدة بمزيد من التفاصيل في الدرس القادم. في الوقت الراهن يمكن أن نفرض أن هذه العقدة تنشئ فقط رسالة جديدة مع موضوع الايميل و نص محتوى الرسالة و التي يمكن ارسالها عن طريق البريد الالكتروني .

حالما يتم إنشاء هذه الرسالة في العقدة function ، يتم إرسالها من خلال التدفق إلى العقدة الأخيرة و هي عقدة البريد الإلكتروني (Email Node) . هذه العقدة تأخذ محتوى اي رسالة واردة و تقوم بإرسالها إلى البريد الالكتروني الذي تم تحديده/اعداده عندما يتم تهيئة العقدة.

اذا التدفق بسيط جدا يستخدم العقد الـ Node-RED للتحقق من الطقس المحل الخاص بك . و اذا كان الطقس معتدل ، ولك تكن قد مارست الكثير من المشي اليوم، يرسل لك رسالة تذكير بالبريد الإلكتروني.

على الرغم من أن المثال يبدو بسيطا جدا، مجرد التحقق من الطقس و ربط بجهاز fitbit الخاص بك، إلا انه يمكنك استخدام التدفقات المماثلة للتحقق من حالة الأجهزة المنزلية،و الأجهزة في المصنع، و ما إلى ذلك.

في مثالنا الأخير ، سوف يبين كيفية إنشاء خدمة ويب بسيطة تستجيب لمتصفح بإرسال طلب HTTP عن طريق الاستعلام عن بعض البيانات في العالم ، ومن ثم يسلم صفحة ويب مع تمثيل رسومي لتلك البيانات الراجعة إلى المتصفح .

بناء خدمة ويب بسيطة باستخدام عقد HTTP المدمجة في Node-RED

في هذا المثال سنقوم بإنشاء خدمة ويب بسيطة تسمح لك الاستعلام عن احصائيات fitibit الخاصة بك لهذا اليوم، وذلك بإستخدام الرسم البياني ، ليتم عرض عدد السعرات الحرارية التي تم حرقها من خلال ممارسة الرياضة.

في التدفق المبين في الشكل أدناه ، يتكون التدفق من 4 عقد. العقد الأولى و الأخيرة هي عقد إدخال و إخراج/انتاج HTTP و التي تعمل معا للاستماع لطلبات HTTP و إرسال ردود HTTP .هذا التدفق يستمع لطلب HTTP من اي مصدر، ولكن دعونا نفترض انه المتصفح العادي هنا. عند وصوله، فإنه يستعلم عن احصائيات fitbit الخاصه بك من fitbit.com ، ثم يستخدم عقدة template لبناء صفحة HTTP ، ويمرر ذلك إلى عقدة انتاج HTTP التي ترسل صفحة الويب إلى المتصفح.

عندما يأتي طلب HTTP ، تقوم عقدة ادخال HTTP بإنشاء رسالة لتعطي اشارة إلى عقدة fitbit و التي تمثل العقدة التالية في التدفق. عقدة fitbit تحصل على الإحصاءات الحالية للمستخدم ، ومن ثم تمرر هذه الاحصائيات كرسالة إلى العقدة التالية عقدة template . عقدة HTML template هي عقدة أخرى بنيت في Node-RED ، و التي تشبه عقدة function ، حيث تسمح لك ببناء تعليمات برمجية . فبدلا من استخدام الجافا سكريبت (javaScript) مثل عقدة function ، فإن عقدة template تعمل مع نص مثل HTML .

يظهر الكود HTML في عقدة template ادناه، حتى تتمكن من رؤية ما يحدث. ومع ذلك، ليس هناك حاجة لدراسة ذلك بالتفاصيل إذا كنت لا ترغب ذلك. في الأساس، عقدة template تأخذ بيانات استهلاك السعرات الحرارية من fitbit باستخدام مكتبة الرسوم البيانية ، تنشئ ملف HTML بسيط من شأنه أن تخبر المتصفح المتلقي لاستخدام مكتبة الرسوم البيانية لعرض الرسم البياني الخاص بغستهلاك السعرات الحرارية .

```
<!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"></script>

<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>
```

وبمجرد أن عقدة template تقوم بإنشاء ملف HTML ، فإنه يمر كرسالة إلى العقدة الأخيرة ، وهي عقدة استجابة HTTP. هذه العقدة تحزم HTML كرد HTTP و التي ترسل إلى المتصفح .

ثم سيظهر للمستخدم رسم بياني بسيط يوضح استهلاك السعرات الحرارية لهذا اليوم ، كما هو بالشكل ادناه_ جميعها بنيت بواسطة تدفق Node-RED بسيط .

تاريخ الـ Node-RED :

كما بينت الأمثلة السابقة، Node-RED هي أداة قوية لبناء تطبيقات و خدمات انترنت الأشي. وأثر نشأتها هي الحاجة لبناء نماذج سريعة لتطبيقات انترنت الأشياء وتم إنشاء Node-RED كمشروع مفتوح المصدر من قبل مجموعة IBM . حيث قامو بإنشاء Node-RED في البداية كأداة لأنفسهم وهم يعملون على مشاريع IOT .

واطلقت النسخة الأولية من Node-RED كمشروع مفتوح في أواخر عام 2013 و تزايدت مجموعة المستخدمين و المطورين خلال عام 2014. وقد شهدت اعتماد كبير من قبل الصناع ، التجريبيون و عدد من الشريكات الكبيرة و الصغيرة حيث قاموا بإستخدامها لتلبيه حاجاتهم .

عندما أنشألت مجموعة IBM الـ Node-RED ، فإنها كانت تركز في الغالب على انترنت الأشياء، أي بمعنى توصل الأجهزة للمعالجات او العكس. كأداة لتطويرالتطبيقات السريعة لإنترنت الأشياء ، الـ Node-RED على حد سواء قوية و مرنة. وقوتها تأتي من مزيج من اثنين من العوامل :

ـالـ Node-RED هي مثال لنموذج البرمجة القائمة على التدفقـ الرسائل تمثل الأحداث التي تتدفق بين العقد ثم يتم معالجتها لتؤدي إلى نتائج.نموذج البرمجة القائم على التدفق هو رسم تفصيلي واضح وجيد لتطبيقات انترنت الأشياء النموذجية.Node-RED تقوم بحزم أحداث العالم الحقيقي كرسائل تقدم نموذجا بسيط و موحد للأحداث و التي تتدفق بين العقد التي تشكل التدفقات .
– مجموعة العقد المدمجة هي الجزء الثاني من قصة الـ Node-RED .فمن خلال بناء مجموعة من عقد الإدخال و الإخراج القوية . و التي تخفي الكثير من تعقيدات التفاعل مع العالم الحقيقي، Node-RED تتيح للمطورين بناء لبنات قوية لتمكينهم من وضعهم معا في تدفق لتحقيق الكثير ، دون الحاجة إلى قلق حول تفاصيل البرمجة.

هذان العاملان جعلو من Node-RED أداة قوية لتطوير تطبيقات انترنت الأشياء .