

Bluemix Hands-On

Web APIを活用したアプリを動かす

Kifumi Numata
University Relations, IBM Japan



Node-REDでWeb APIを活用したアプリを動かす

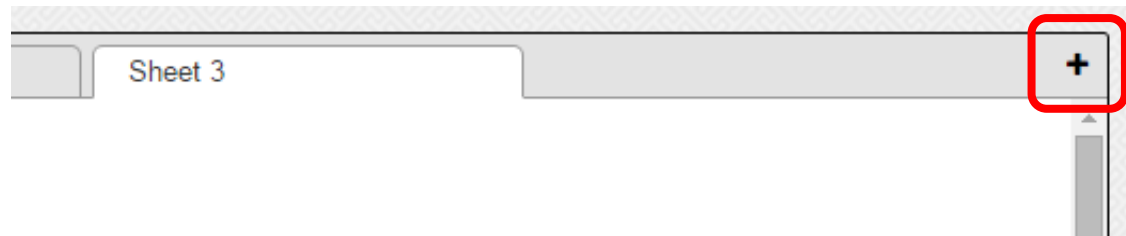
- Node-REDは作成したフローをJSON形式の文字列としてexport/importすることができます。
- あらかじめ作成済みの処理フローをNode-REDに取り込んでみましょう。
- Githubにあるサンプルアプリのテキストファイル（5_WebAPI_成.txt）の文字列をすべて選択→コピーします。

```
[{"id":"9c253959.9a24c8","type":"websocket
out","z":"26233161.4aac26","name":"","server":"","8431bf2d.ad5f6","x":636,"y":191.36666870117188,"wires":[{}],{"id":"4090a6f0.f2ea58","type":"xml","z":"26233161.4aa
c26","name":"","attr":{"$":"chr":"_","x":457.5,"y":192.36666870117188,"wires":[{"id":"9c253959.9a24c8","7a50f0a0.3f88b8"}]},{"id":"7a50f0a0.3f88b8","type":"debug","z":"2
6233161.4aac26","name":"","active":true,"console":false,"complete":false,"x":585.5,"y":257.3666687011719,"wires":[{"id":"b044e7d0.181c2","type":"http
request","z":"26233161.4aac26","name":"リクナビAPI呼び出し
","method":"GET","url":"http://web.service.recruit.co.jp/shingaku/school/v2/?code=SC000332&key=43ba47966ce3abea","tls":"","x":285,"y":192.36666870117188,"wir
es":[{"id":"4090a6f0.f2ea58"}]},{"id":"45d0d2da.da89ac","type":"websocket
in","z":"26233161.4aac26","name":"","server":"","8431bf2d.ad5f6","x":85,"y":192.36666870117188,"wires":[{"id":"fd8f0780.3dda","type":"http
response","z":"26233161.4aac26","name":"","x":451,"y":98.36666870117188,"wires":[{"id":"49b83e5e.2a6708","type":"template","z":"26233161.4aac26","name":"
表示内容の定義","field":"payload","fieldType":"msg","syntax":"mustache","template":"<!DOCTYPE html>¥n<html>¥n <head>¥n <meta name=¥"viewport¥"
content=¥"initial-scale=1.0, user-scalable=no¥">¥n <meta charset=¥"utf-8¥">¥n <title>成城大学はここ！</title>¥n <style>¥n html, body, #map-canvas {¥n
height: 92%;¥n margin: 5px;¥n padding: 0px;¥n }¥n </style>¥n <script src=¥"https://maps.googleapis.com/maps/api/js?v=3.exp¥"></script>¥n
<script>¥n¥n¥nfunction initialize() {¥n¥n¥n var myLatLng = new google.maps.LatLng(35.64,139.6);¥n¥n¥n var mapOptions = {¥n¥n¥n zoom: 13,¥n¥n¥n center:
myLatLng¥n¥n¥n };¥n¥n¥n var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);¥n¥n¥n ¥n¥n¥n var loc = window.location;¥n
if (loc.protocol === ¥"https:¥") {¥n newUri = ¥"wss:¥";¥n } else {¥n newUri = ¥"ws:¥";¥n }¥n newUri += ¥"//¥" + loc.host +
¥"/ws/stations¥";¥n¥n¥n ¥n¥n¥n var sock = new WebSocket(newUri);¥n¥n¥n sock.onopen = function(){¥n¥n¥n console.log(¥"Connected
websocket¥");¥n¥n¥n console.log(¥"Sending ping. ¥");¥n¥n¥n sock.send(¥"Ping¥");¥n¥n¥n console.log(¥"Ping sent. ¥");¥n¥n¥n };¥n¥n¥n sock.onerror =
function(){ console.log(¥"Websocket error¥"); };¥n¥n¥n sock.onmessage = function(evt){¥n¥n¥n var schoolData = JSON.parse(evt.data);¥n¥n¥n var stations =
schoolData.results.school[0].campus;¥n¥n¥n for(var i = 0; i < stations.length; i++) {¥n¥n¥n var camp = stations[i];¥n var marker = new
google.maps.Marker({¥n¥n¥n position: new google.maps.LatLng(camp.latitude[0],camp.longitude[0]),¥n¥n¥n map: map,¥n¥n¥n animation:
google.maps.Animation.DROP,¥n¥n¥n title: camp.address[0]¥n¥n¥n });¥n¥n¥n
¥n¥n¥n¥n¥n¥n };¥n¥n¥n¥n¥n¥n google.maps.event.addDomListener(window, 'load', initialize);¥n¥n </script>¥n </head>¥n <body>¥n <div id=¥"map-
canvas¥"><div>¥n <font size=¥"6¥">成城大学はここ！</font>¥n <br>powered by IBM Bluemix and Node-RED¥n <div align=¥"right¥"> ¥n <a
href=¥"http://web.service.recruit.co.jp/¥"><img src=¥"http://web.service.recruit.co.jp/banner/shingaku-m.gif¥" alt=¥"リクナビ進学 Webサービス¥" width=¥"88¥"
height=¥"35¥" border=¥"0¥" title=¥"リクナビ進学 Webサービス¥"></a>¥n </div> ¥n
</body>¥n</html>","x":289,"y":98.36666870117188,"wires":[{"id":"fd8f0780.3dda"}]},{"id":"b00beb7.e8c1718","type":"http
in","z":"26233161.4aac26","name":"","url":"/schoolmap","method":"get","x":102,"y":98.36666870117188,"wires":[{"id":"49b83e5e.2a6708"}]},{"id":"8431bf2d.ad5f6","type":
"websocket-listener","path":"/ws/stations","whomsg":"false"]}
```

Node-REDでWeb APIを活用したアプリを動かす

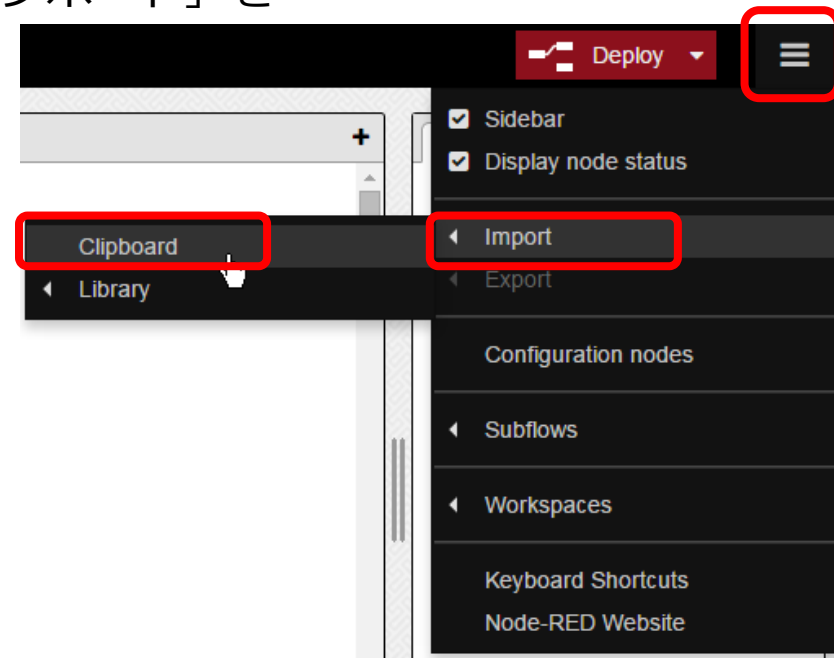
■ Node-REDでシートを追加します。

上部の「+」（プラス）ボタンをクリックして、Sheet 2 を開きます。



■ Sheet 2 に先程コピーした処理フローを取り込みます。

上部右端の三本線のアイコンをクリックし、表示されたメニューから「import」→「Clipboard」または、「読み込み」→「クリップボード」を選択します。



Node-REDでWeb APIを活用したアプリを動かす

- 「Import nodes」ウィンドウが表示されるので、
先程クリップボードにコピーしたテキストをペースト
して、「OK」または「読み込み」ボタンをクリックします。

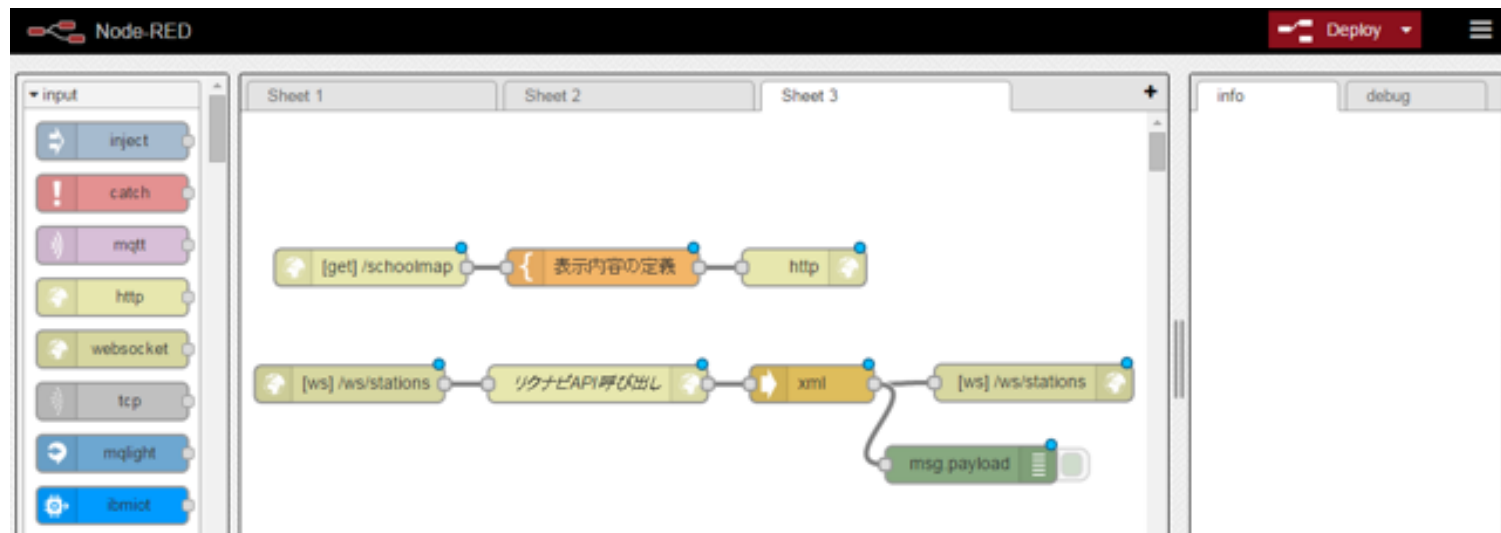
Import nodes

```
{
  "id": "6c9867b7.936798",
  "type": "debug",
  "name": "",
  "active": true,
  "console": "false",
  "complete": "false",
  "x": 582,
  "y": 333,
  "z": "fd69388c.0296c8",
  "wires": []
}, {
  "id": "d13507f1.2ecaf8",
  "type": "websocket",
  "in": "",
  "name": "",
  "server": "13257eec.669068",
  "x": 81.5,
  "y": 268,
  "z": "fd69388c.0296c8",
  "wires": [
    [
      "e9e6c158.16194"
    ]
  ]
}
```

Ok

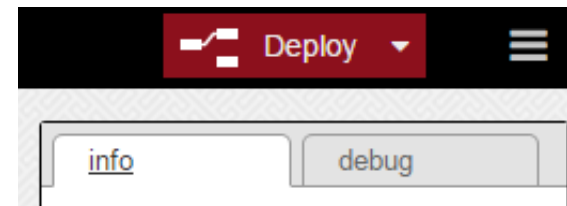
Cancel

- 処理フローが取り込まれますので、キャンバスの適当な場所をクリックして配置します。



Node-REDでWeb APIを活用したアプリを動かす

- このアプリを動かしてみよう。
右上の「Deploy」ボタンをクリックします。



- 画面上部に「Successfully Deployed」の文字が表示されればOKです。
- 実行結果を確認してみましょう。

Webブラウザの新しいタブを開いて、Node-REDのホスト名の末尾に「/schoolmap」というパスを付加したURLにアクセスします。

(例) <https://xxx.mybluemix.net/red/#> でNode-REDを開いている場合、



<https://xxx.mybluemix.net/schoolmap> にアクセスします。

Node-REDでWeb APIを活用したアプリを動かす

- Node-REDで作ったサンプルアプリ「成城大学はココ！」が表示されればOKです。



地図上の成城大学のキャンパスにマーカーが表示され、マウスポインタを合わせると大学の住所が表示されます。

Node-REDでWeb APIを活用したアプリを動かす

- debug画面の「▶」をクリックして、リクナビ進学APIを呼び出した結果が表示されていることを確認してください。
Webブラウザに表示していないものも含め、様々な情報が取得できています。

The screenshot shows the Node-RED web interface. On the left, the 'Inputs' palette contains various nodes like inject, catch, status, link, mqtt, http, websocket, tcp, mqlight, and ibmiot. The main workspace displays a flow with three parts: 1) A 'get' node for '/schoolmap' connected to a function node '表示内容の定義', which is then connected to an 'http' node. 2) A 'ws' node for '/ws/stations' (labeled 'connected 1') connected to a function node 'リクナビAPI呼び出し', which is then connected to an 'xml' node. 3) Another 'ws' node for '/ws/stations' (labeled 'connected 1') connected to a 'msg.payload' node. The right-hand 'debug' console is open, showing a log entry for '017/5/18 9:53:22 node: 7a50f0a0.3f88b8'. The log content is a JSON object:

```
msg.payload: Object
  results: Object
    $: Object
      api_version: array[1]
      results_available: array[1]
      results_returned: array[1]
      results_start: array[1]
    school: array[1]
      0: Object
        code: array[1]
          0: "SC000332"
        name: array[1]
          0: "成城大学"
        kana: array[1]
          0: "セイジョウダイガク"
        title: array[1]
          0: "実践的な少人数教育で個性を磨き、社会の最前線で活躍できる人材に"
```

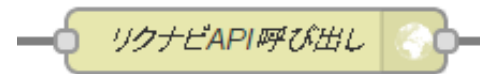
このようにクラウドで公開されている外部のAPIを呼び出すことで、アプリの機能を拡張できます。

Node-REDでWeb APIを活用したアプリを動かす

■ 試してみましょう

リクナビ進学APIに対してリクエストする内容を変えてみましょう。

■ 「リクナビAPI呼び出し」ノードをダブルクリックして構成画面を開きます。



■ URL欄の「code= SC000332」の部分を「code= SC000555」と変更して「OK」または「完了」をクリック。

A screenshot of the 'Edit http request node' dialog in Node-RED. The 'URL' field is highlighted with a red rectangle and contains the text 'http://webservice.recruit.co.jp/shingaku/sct'. To the right of the dialog, a preview of the URL is shown: 'gaku/school/v2/?code=SC000163&key=43ba4796'. At the bottom of the dialog, the 'Ok' button is highlighted with a red rectangle. The 'Method' is set to 'GET', 'Return' is empty, and 'Name' is 'リクナビAPI呼び出し'. There is a checkbox for 'Use basic authentication ?' which is unchecked.

■ デプロイして結果を見てみましょう。

右上の「Deploy」ボタンをクリックします。

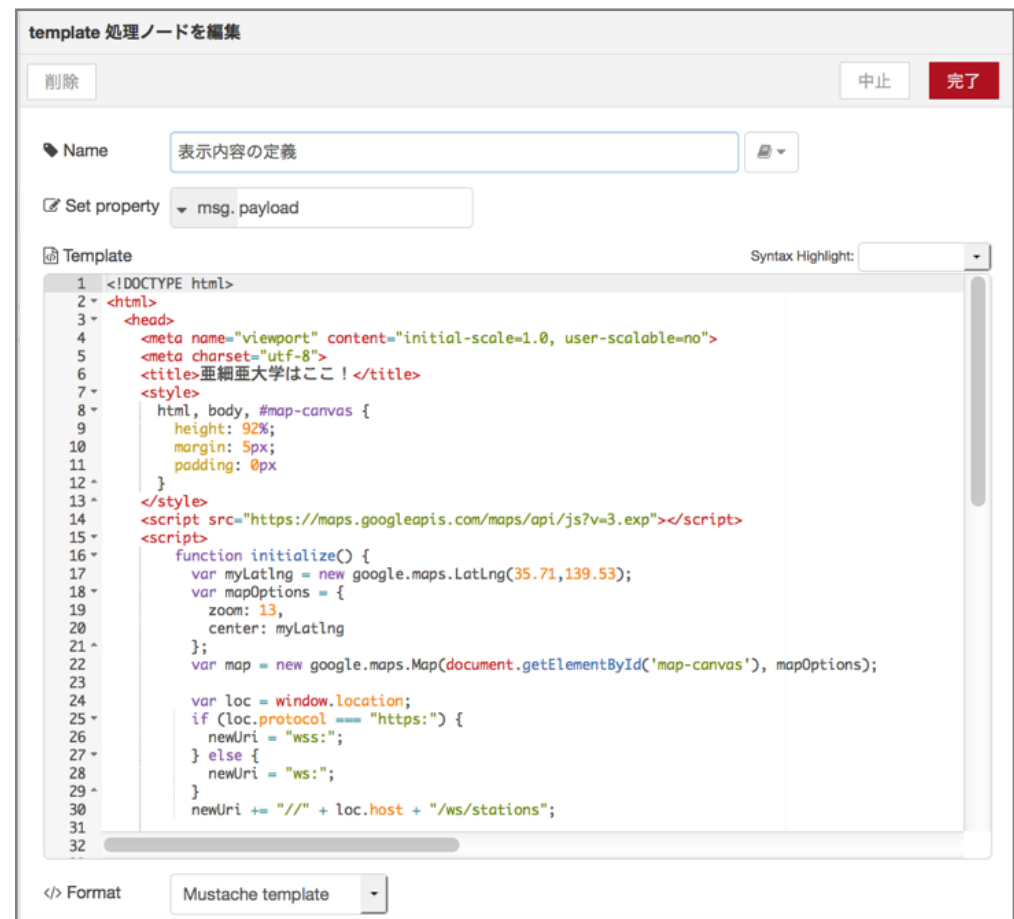
「成城大学はココ！」をWebブラウザで再読み込みすると、何が変わるでしょうか？

Node-REDでWeb APIを活用したアプリを動かす

■応用： 試してみましょう

「成城大学はココ！」の表記を変えてみましょう。

「表示内容の定義」ノードをダブルクリックして、どこを編集すればよいか試してみましょう。



API活用によるアプリ開発とIBM Bluemix

- 「成城大学はココ！」アプリはBluemix環境のNode-RED上で稼動しています。

地図はGoogle Maps APIを、大学の住所および緯度経度の情報はリクナビ進学APIを呼び出して統合し、Webブラウザ上に表示しています。

