



+

Node-RED

Flow-based programming for the Internet of Things



OpenSource協議会-IBM i

Ver. 1.0

2019-05-24

IBM i と連携する Node.js と Node-RED で  
アプリ構築を体験しよう

User & IBM **NEXT 2018**



# アジェンダ

# アジェンダ

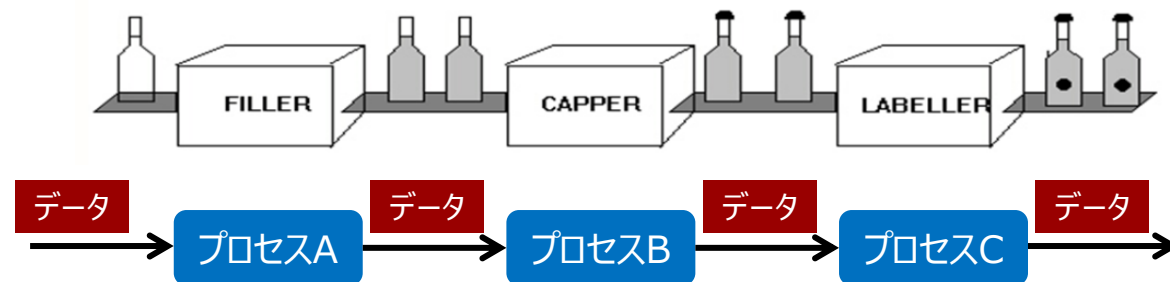
1. Node-RED の前段として
2. Node-RED ハンズオンのご紹介

# Node-RED 前段

# Node-RED

## □ フローベースド・プログラミング・ツール

- アプリケーションの動作をブラックボックスのネットワークによって表す手法



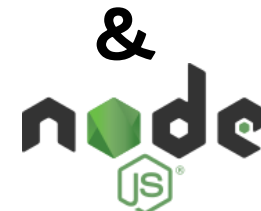
## □ つなぐためのオープンソース・ソフトウェア

- ハードウェアデバイス
- **API**
- **Webサービス (Web API)**
- etc ...

## □ API 連携を簡単に実現するツール

Node-RED

Flow-based programming for the Internet of Things



# API (Application Programming Interface)

## □ ソフトウェアから別のソフトウェアの“機能”を呼び出す仕組み

- 古くからシステム開発の現場で使われている

## □ 特徴

- 同じ機能を簡単に再利用可能
- 実装の詳細を隠蔽 (知る必要がない)
- プログラムの機能を呼び出し、実行結果を受け取る



# API の変化

## □ API - 位置づけの変化

### 1. 再利用の効率化：ソフトウェア開発

- ・ サブルーチン
- ・ 外部プログラム（外サブ）

### 2. OS 提供 API の利用：プラットフォーム

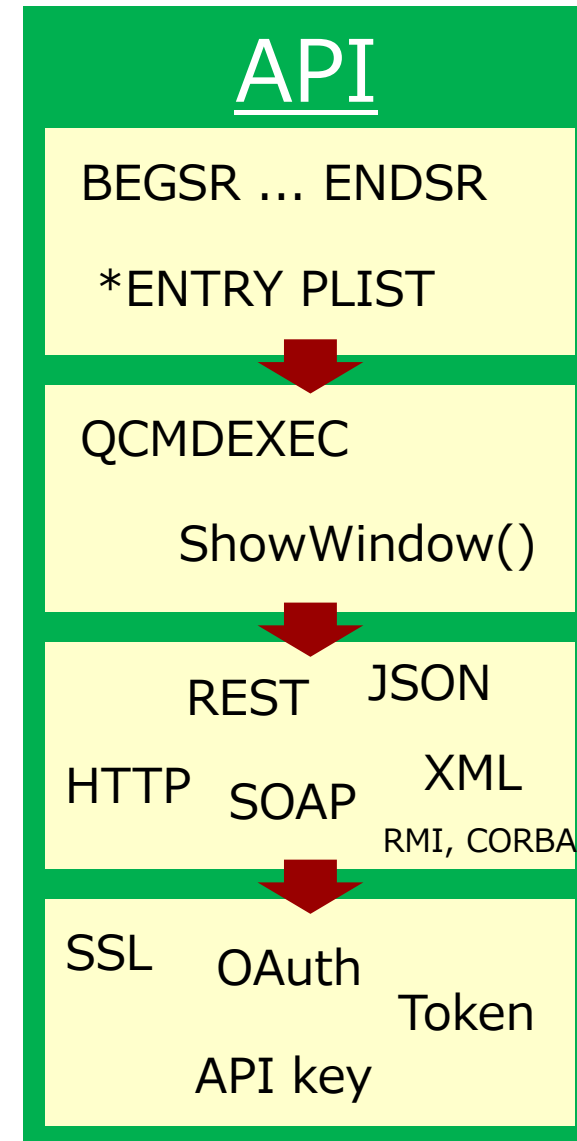
- ・ IBM i API、
- ・ Windows API

### 3. WebAPI による相互利用：インターネット

- ・ 汎用性の拡大（外部から利用）
- ・ 社内利用の充実

### 4. 経済のオープン化：オープン API

- ・ 一般に公開し利用してもらう
- ・ セキュリティーの実装



# オープンAPI

## □ API を他の企業等に公開する事を指す

- 価値のある資産やサービスへのアクセス方法を外部に公開する
  - 接続可能性を示すことが、自社の強みを最大化することに結びつく考え方
- インターフェースを公開し、同業他社に共有してマーケットを広げる
  - その上で裏側のサービス品質を高めて差別化するという戦略





# オープンAPI

## □ オープンAPI を後押しする技術的背景

### – クラウド化：外部連携の容易化

- クラウドで公開する事で、セキュリティーを強固に

### – アナリティクス：高度な分析、判断

- 知見、AI の利用により創造した価値を提供

### – モバイル化：タイムリーな連携

- デバイスの多様化により、時間/場所を問わず連携が可能に

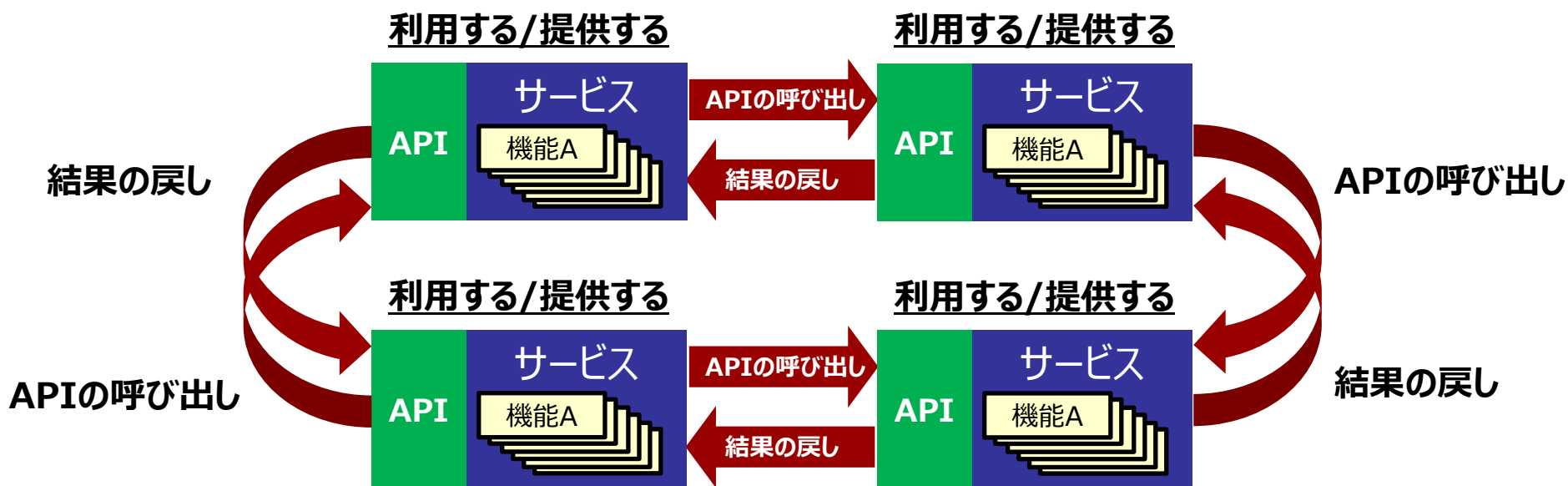
### – ソーシャルメディア：大規模コミュニティから発信される情報利用

- 世論、動向の活用



## □ API を活用した新たな経済圏

- APIの公開で、他社サービスも活用し広がっていく商圈（経済圏）
  - APIがビジネスとビジネスをつなぎ、企業同士がお互いの強みを利用して、新たな価値を創出する動き
  - APIがビジネスを広げるために活用されてきている
  - APIを公開する企業や政府もグローバルで増えつつある

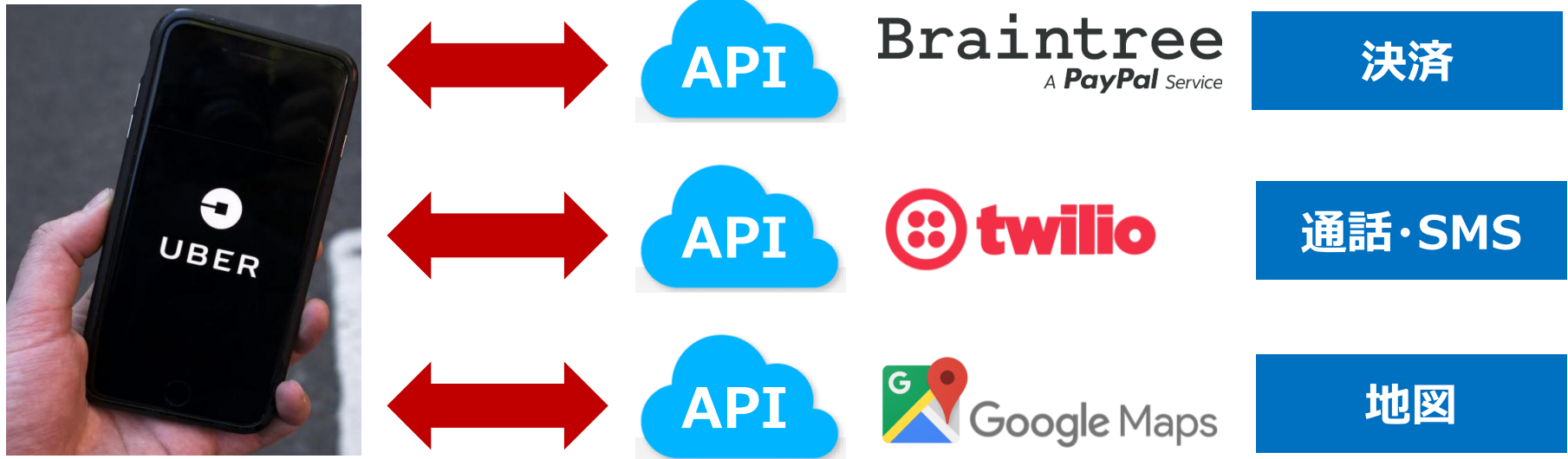


# API エコノミーの事例

## □ API を活用した新たな経済圏

### - Ubar : 配車アプリ

- Uberは、「地図」、「通話・SMS」、「決済」などのAPIを組み合わせることで、「マッチング」という自らのコア機能の開発に専念し、サービスを迅速に立ち上げることができた。



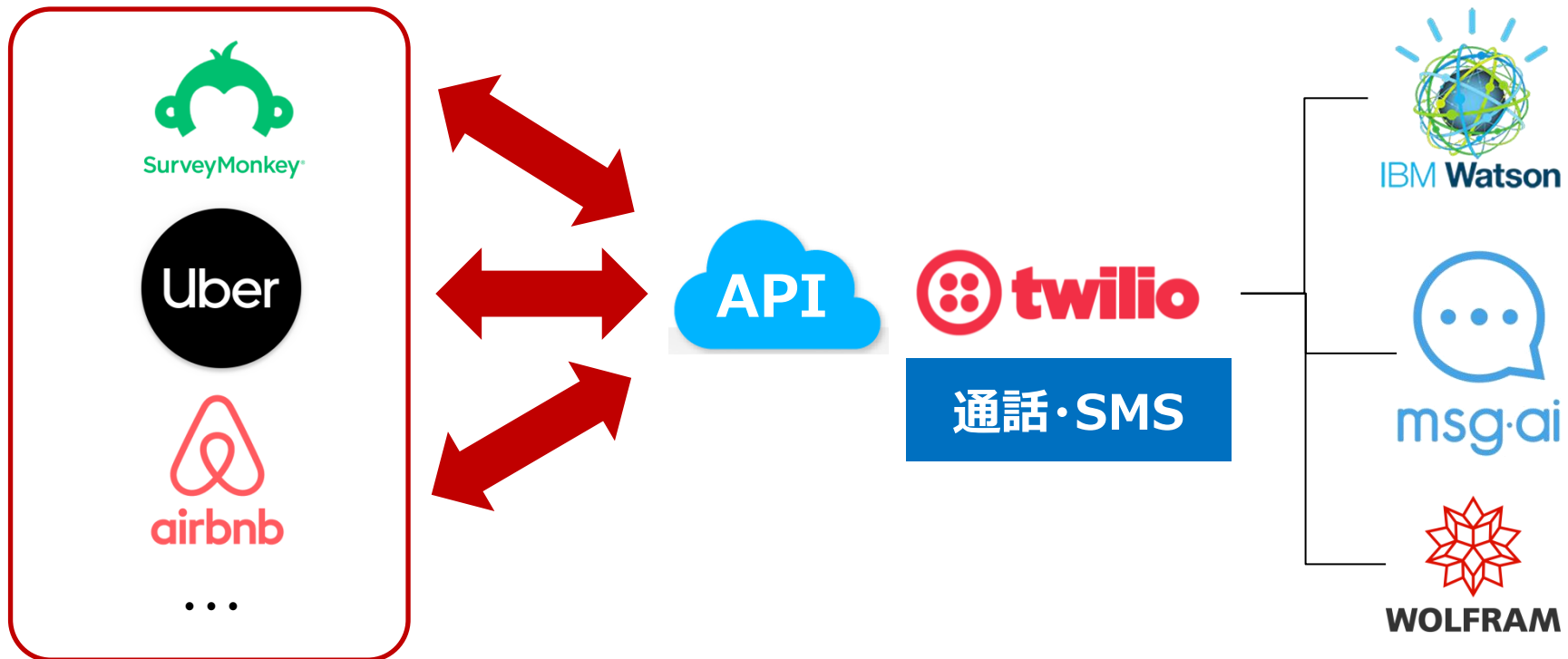
※ APIサービスの活用事例です。Node-RED の事例ではありません

# API エコノミーの事例

## □ API を活用した新たな経済圏

### - Twilio : 通話・SMS

- 他社がAPIの機能を拡張してくれる上に、他社のアドオンの販売額の25%がTwilioの取り分になるという仕組み



※ APIサービスの活用事例です。Node-RED の事例ではありません

# デジタルトランスフォーメーション(DX) を API エコノミーで実現

APIエコノミーでデジタルトランスフォーメーション

https://www.ibm.com/jp-ja/cloud/api-economy

IBM | Cloud 製品 ソリューション 料金設定 パートナー 資料 サポート Cloud登録/ログイン Search

## デジタル・トランスフォーメーションをAPIエコノミーで実現

データを収益化して新たな収益源を創出するリーダー企業が  
続々登場

「APIエコノミー最前線」資料をダウンロードする

- ↓ 導入事例
- ↓ API公開のメリット
- ↓ 業界別ユースケース
- ↓ API管理とは
- ↓ リソースを活用する
- ↓ ソリューション

### API公開成功事例

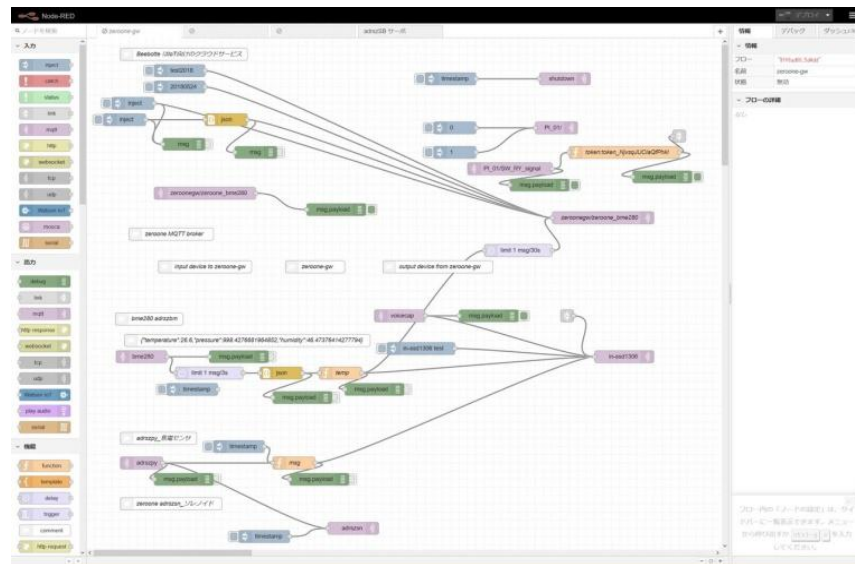
ご相談ください

# NEXT 2018 ハンズオン

# Node-RED 概説

## □ Node-RED ?

- ハードウェアデバイス、API、オンラインサービスをつなげるためのプログラミングツール
- IBM 英国ハーズレイ研究所が開発したソフトウェア
- オープンソースで提供されている
- Node.js で作られている
- 開発は「ブラウザ」で行う
- 基本はマウスによる操作
- JavaScript でコーディングも可能



Node-RED

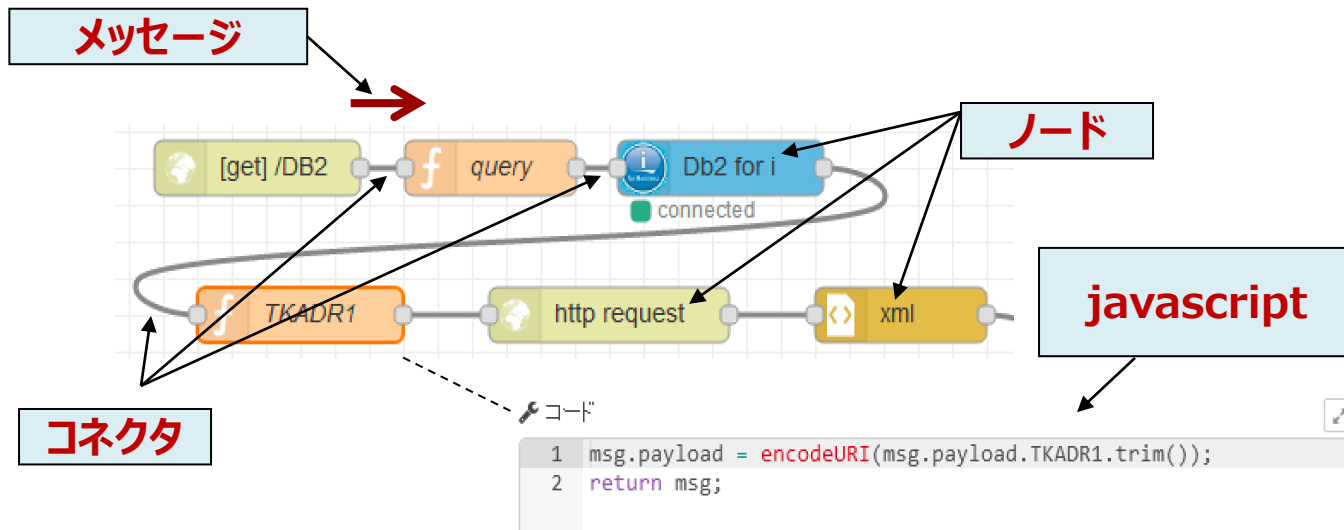
Flow-based programming for the Internet of Things



# Node-RED 概説

## □ 開発の概要

- ノードと呼ばれる部品をマウス操作でエディタに配置する
- ノードとノードをマウス操作によりコネクタで接続する
- 接続したノードからノードへ「**メッセージ**」を渡し処理を行う
- メッセージにはJsonが利用される
- 細かな処理は javascript でコーディングできる





# Node-RED 概説

## □ 開発環境の概要

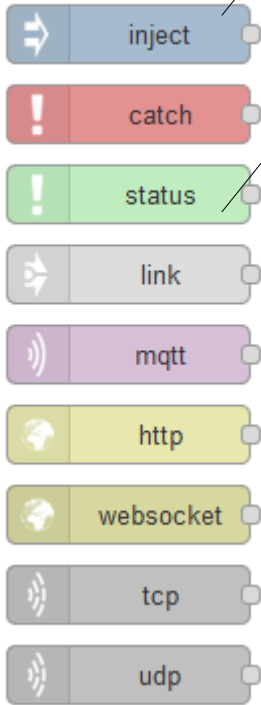
The screenshot shows the Node-RED interface with several components highlighted by red boxes and labeled with callouts:

- デプロイボタン** (Deploy Button): Located in the top right corner of the interface.
- メニュー** (Menu): Located in the top right corner, next to the deploy button.
- パレット** (Palette): The left sidebar containing various nodes categorized into '出力' (Output) and '機能' (Function).
- フローエディタ** (Flow Editor): The central workspace where the flow is built, showing nodes like 'Hello Worlda', 'msg.payload', '[get] /DB2', 'query', 'Db2 for i', 'TKADR1', 'http request', 'xml', and another 'http request' node.
- コンソール** (Console): The right sidebar showing the debug console output, including a timestamp, node ID, and the payload of the message.

# Node-RED 概説

## ノードの種類

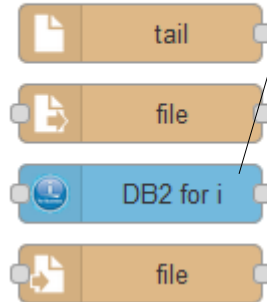
### 入力



Inject : フローの開始点  
手動, スケジュールなどフローの  
開始が行える

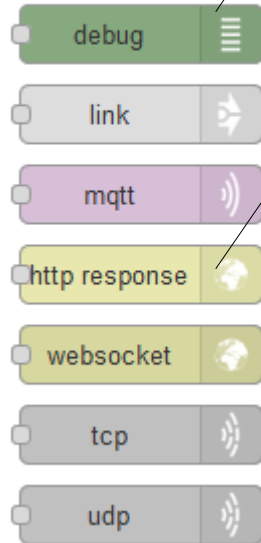
http : フローの開始点  
http 要求を開始点として扱える

### ストレージ



DB2 for i : IBM i データ操作  
IBM i DB2 との接続を行う

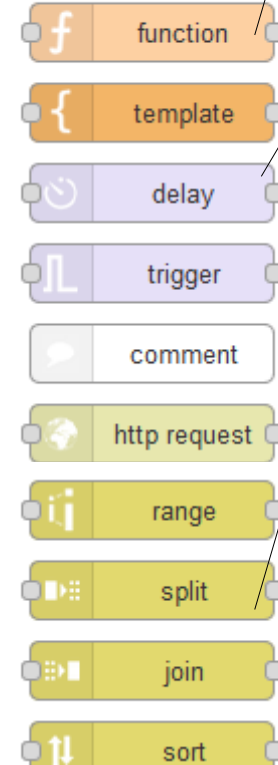
### 出力



debug : デバッグ出力  
受け取ったメッセージを  
コンソールに出力する

http response :  
受け取ったメッセージを  
コンソールに出力する

### 機能



function : データの加工等  
JavaScript によりデータの加工  
等を行う

delay : 遅延、流量制限  
遅延時間を設定し次ノードの処理  
集中を抑止する

split : データの分割  
受け取ったデータを分割し次のノード  
へ出力する

## □ 課題は3つ

1. hello world
2. IBM i からデータを取り出してみる
3. オープン API と接続

## □ ハンズオンの時間

- 35~40分
- 全員完了 (15名)

# ハンズオン概説 1

## □ hallo world で操作方法を理解する

The screenshot shows the Node-RED web interface in a browser. The main workspace contains a flow with two nodes: 'Hello World' (orange) and 'msg.payload' (green). A yellow callout box points to the 'Hello World' node with the text: "Hello World" をメッセージ出力. Another yellow callout box points to the 'msg.payload' node with the text: "Hello World" メッセージを受け取り、コンソールへ出力. On the right, the 'デバッグ' (Debug) console shows the output: 2019/5/23 10:51:07 node: fa2504cd.b024b8 msg.payload : string[11] "Hello World".

inject ノードを編集

削除 中止 完了

プロパティ

ペイロード

トピック

出力

debug

# ハンズオン概説 2

## □ IBM i からデータを取り出してみる

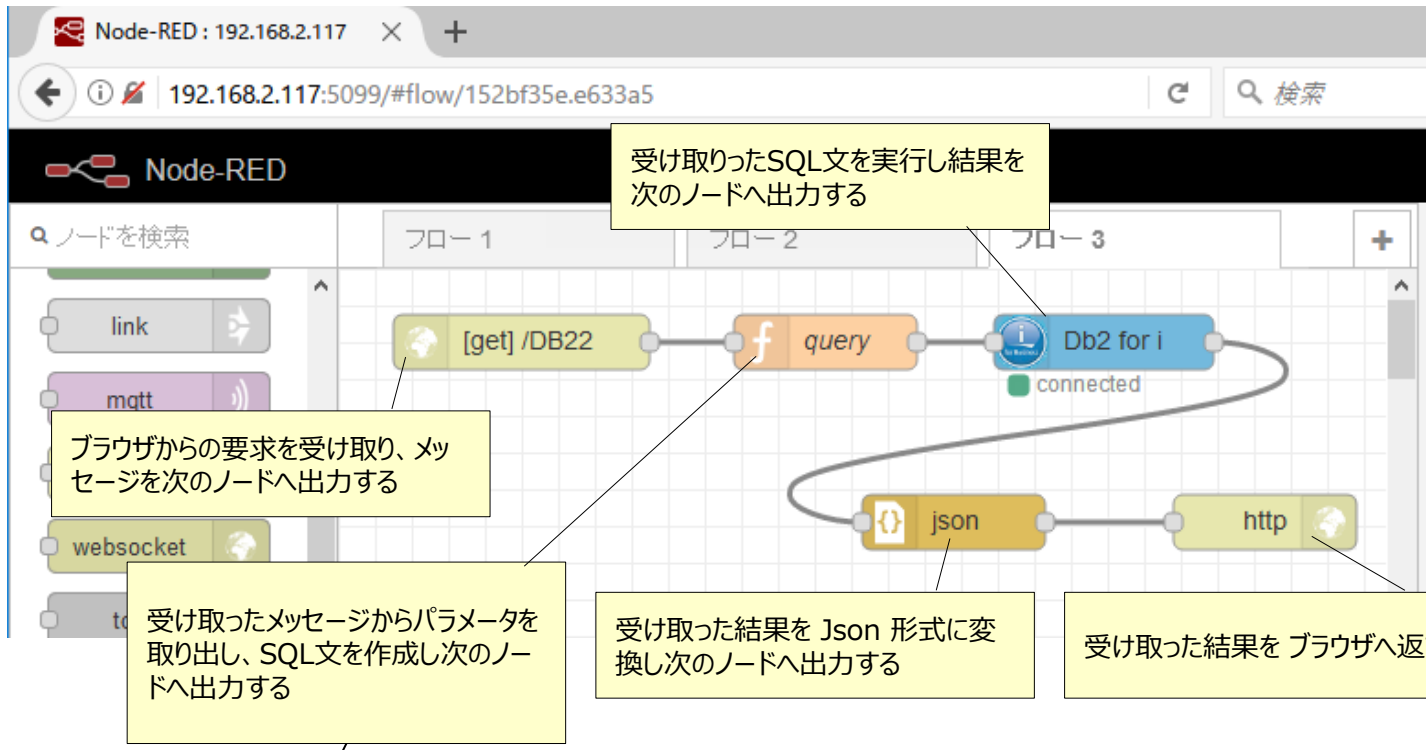


## □ 処理の流れ

- ① ブラウザから要求（パラメータ：得意先名）
- ② 要求された得意先名に該当するデータを取得
- ③ 取得したデータを Json 形式に変換しブラウザに返す

# ハンズオン概説 2

## 結果

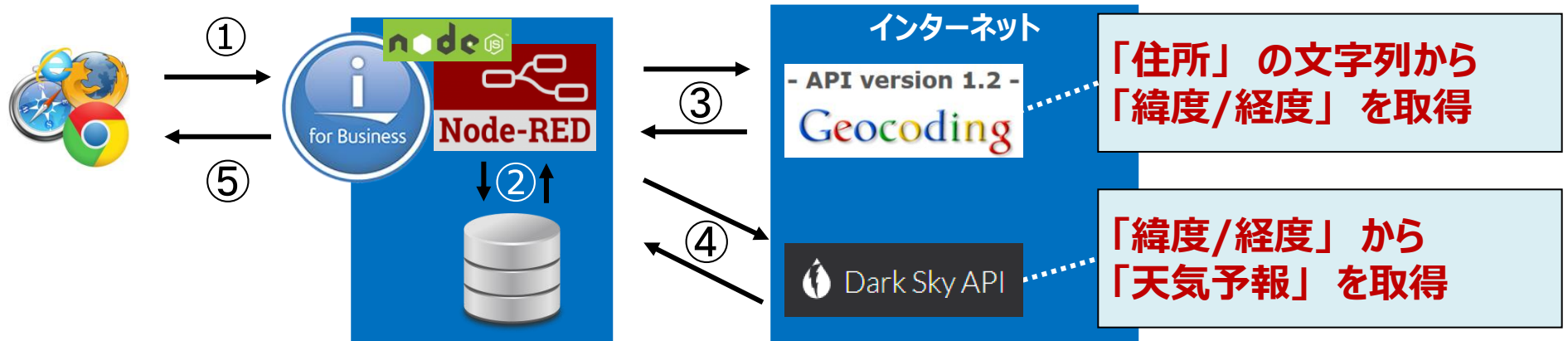


```
var sql = "select * from QEOL.TOKMSP where TKNKJ like '%" + msg.payload.tknakj + "%'";  
msg.payload = sql;  
return msg;
```

```
192.168.2.117:5099/DB22?tknakj= X +  
保護されてい | 192.16  
アプリ IBM i + Dev T&Trust  
[  
- {  
  TKBANG: "01030",  
  TKNKJ: "アブリ コパコ",  
  TKNKJ: "相川工業",  
  TKADR1: "東京都世田谷区",  
  TKADR2: "若林4-24",  
  TKTIKU: "08",  
  TKPOST: "154",  
  TKTELE: "03-964-6406",  
  TKGURI: "136200",  
  TKNURI: "243000",  
  TKZURI: "796600",  
  TKUZAN: "110000",  
  TKGEND: "1120000",  
  TKNYUK: "880619",  
  TKSIME: "1",  
},  
- {  
  TKBANG: "01070",  
  TKNKJ: "アブリ カラ",  
  TKNKJ: "相川カメラ",  
  TKADR1: "東京都新宿区",  
  TKADR2: "四谷1-16",  
  TKTIKU: "03",  
  TKPOST: "180",  
  TKTELE: "03-354-3018",  
  TKGURI: "367900",  
  TKNURI: "2011700",  
  TKZURI: "3914500",  
  TKUZAN: "980000",  
  TKGEND: "1000000",  
  TKNYUK: "880611",  
  TKSIME: "1",  
},  
- {  
  TKBANG: "01080",  
  TKNKJ: "アブリ コパコ.K",  
  TKNKJ: "相川広告K.K",  
  TKADR1: "東京都渋谷区",  
  TKADR2: "広尾3-9",  
  TKTIKU: "02",  
  TKPOST: "150",  
  TKTELE: "03-368-6366",  
},  
[1]
```

# ハンズオン概説 3

## □ 外部の API サービスを利用してみる

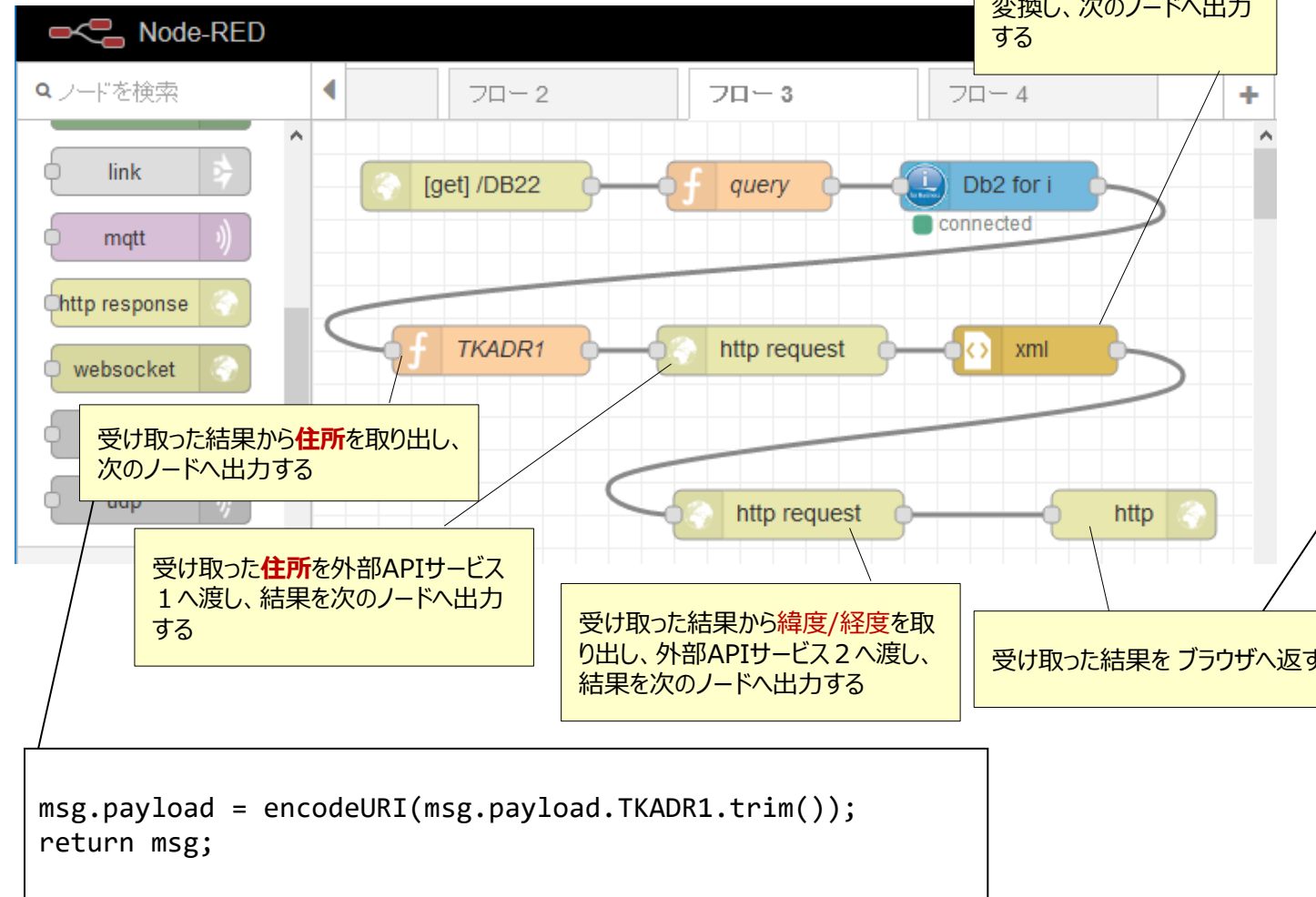


## □ 処理の流れ

- ① ブラウザから要求（パラメータ：得意先名）
- ② 要求された得意先名に該当するデータを取得
- ③ データから住所を取り出し外部APIサービス 1 から緯度/経度を取得
- ④ 取得した緯度/経度で外部APIサービス 2 から天気予報を取得
- ⑤ 結果を Json 形式に変換しブラウザに返す

# ハンズオン概説 3

## □ 結果



192.168.2.117:5099/DB22?tknakj: x +  
← → ↻ ⓘ 保護されていない | 192.168.2...

```
{
  latitude: 35.648562,
  longitude: 139.653292,
  timezone: "Asia/Tokyo",
  - currently: {
    time: 1558581161,
    summary: "Clear",
    icon: "clear-day",
    precipIntensity: 0,
    precipProbability: 0,
    temperature: 75.86,
    apparentTemperature: 75.86,
    dewPoint: 48.35,
    humidity: 0.38,
    pressure: 1009.02,
    windSpeed: 8.11,
    windGust: 8.11,
    windBearing: 139,
    cloudCover: 0.16,
    uvIndex: 9,
    visibility: 6.4,
    ozone: 312.06
  },
  - hourly: {
    summary: "Clear throughout the day.",
    icon: "clear-day",
    - data: [
      - {
        time: 1558580400,
        summary: "Clear",
        icon: "clear-day",
        precipIntensity: 0,
        precipProbability: 0,
        temperature: 75.55,
        apparentTemperature: 75.55,
        dewPoint: 48.52,
        humidity: 0.39,
        pressure: 1009.11,
        windSpeed: 7.89,
        windGust: 7.89,
        windBearing: 138,
        cloudCover: 0.18
```



# 自己紹介

T&T

「優しさ」と「信頼」で  
お客様のビジネスに “Goode Cycle” を。



+  $\alpha$  =

**T & T** Trust  
Tenderness and Trust  
<http://tat.co.jp>

## 受託開発

信頼性においては定評のある IBM i をメインとしたアプリケーション開発を行います。お客様とのコミュニケーションを第一に考え、最適なソリューションをご提案しています。

## 顧客研修

外部研修にて IBM i コースを担当する専任のインストラクターがいます。また、お客様のご要望に応じたオーダーメイドの研修コースの提供も行います。

## 技術開発

IBM i の最新技術だけでなく、お客様にとって必要になるであろうコンピュータ技術を日々蓄積しています。

# ティアンドトラスト株式会社

〒111-0053

東京都台東区浅草橋4-16-4 ウイングイトビル6F

フリーダイヤル : 0120-913-474

代表 : 03-5821-3666