

Atelier Akihabara

Pepper×Watson連携講座①

2017/5/21

Softbank Robotics



Pepperで撮影した写真をWatsonに送って
画像認識をして、Pepperに発話させてみよう！

1.簡単な前置き（自己紹介など）

2.Watsonとは

3.Watsonことはじめ

4.Pepperで写真をとるには

5.ハンズオン(写真をHTTP Post で送信)

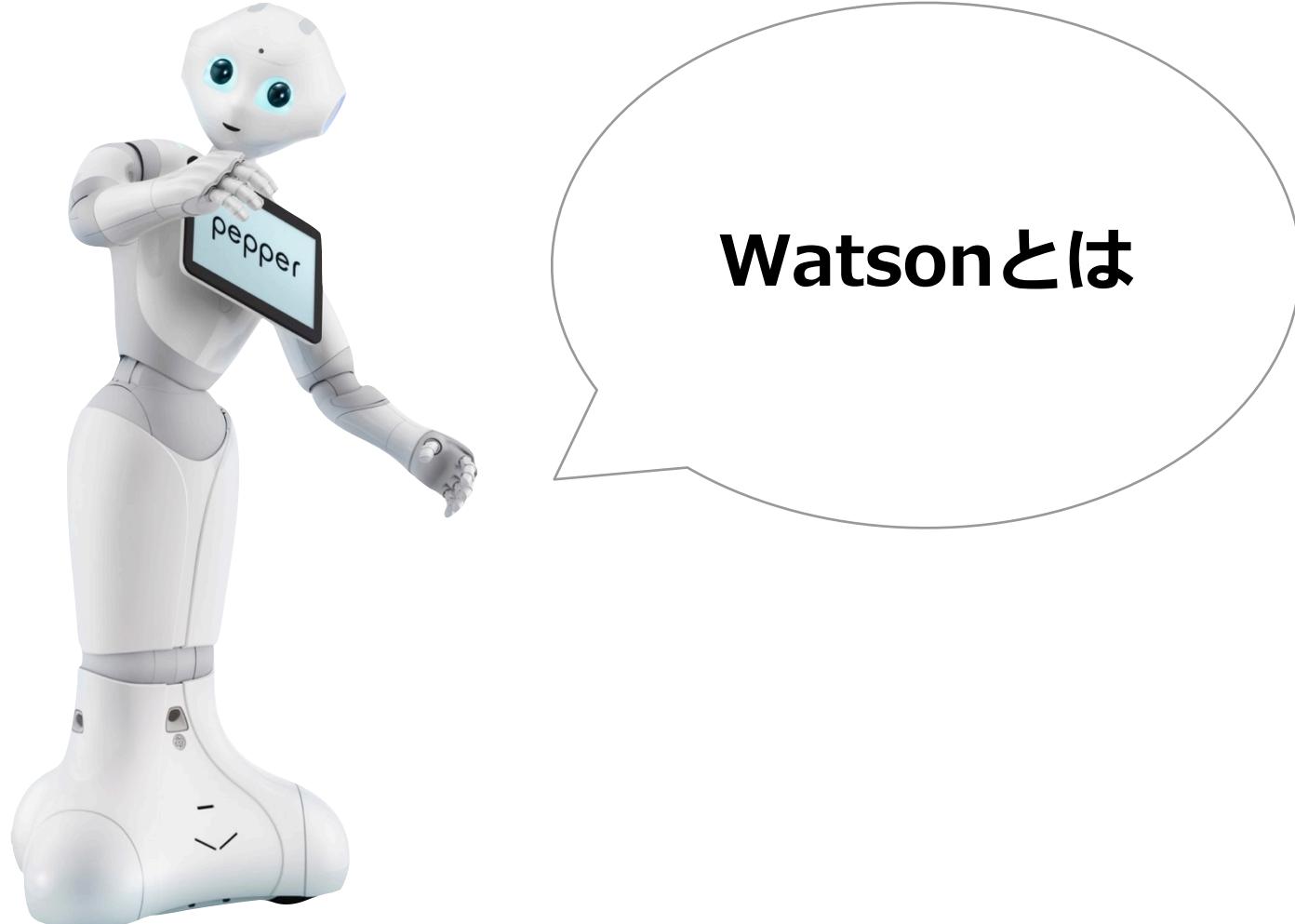
皆様にも 30 秒程度でよいので自己紹介をお願いします！

- ・お名前
- ・ご所属
- ・プログラミング経験

(オプション)

- ・本日の意気込み

一言いただければ幸いです



Watsonについて

IBM社が提供する
『**コグニティブ・コンピューティング・システム**』であり
IBM Cloudの一環。



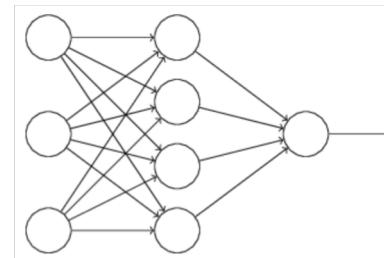
Watsonの基本的な用途



質問対応の自動化



資料からの
自動情報抽出



データ同士の
関係の洞察

→ その裏では**Deep learning**という仕組みが用いられている

Deep Learning とは

- Deep Learning (深層学習)

- 生物の脳機能を模した機械学習手法 Neural Network のなかでも「深い」構造を持った **Deep Neural Network** による機械学習

- 近年の**人工知能**ブームの火付け役

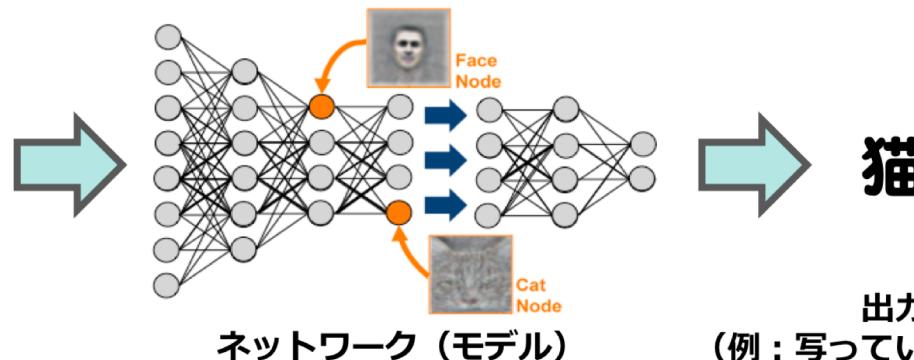
- 例：AlphaGo (囲碁), Ponanza (将棋), 自動運転, Google翻訳, 商品推薦 (Amazon), 音声認識 (Yahoo!), etc.

- 入力情報から出力情報を予測するように**ネットワーク（モデル）**を学習する

- 例：画像から何が写っているかを予測（画像認識）



入力
(例：画像)



ネットワーク（モデル）

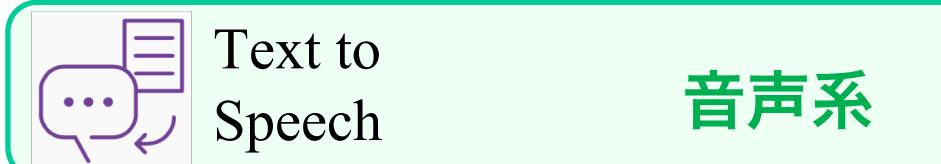
出力
(例：写っている物体名)

Watsonが提供するサービス



- Language
Translator
- Natural Language
Classifier
- Personality
Insights
- Tone Analyzer

言語系



音声系



1. Watson developer cloudでWatsonに直接アクセス

Watsonの機能を利用するためのライブラリ。

Python, cUrl, Node.js, Javaに対応したものがそれぞれ用意されている。

2. REST APIでWatsonに直接アクセス

HTTP通信でサーバとやりとりする方法。

Pythonの場合はrequestsライブラリを用いるのが王道

3. REST APIでNode-redを介してWatsonに間接アクセス

IBM Cloud上でGUIプログラミングツールのNode-redを展開。

Watsonの各機能に対応したノードを用いる

Watsonの機能を利用するためのライブラリ

→ Github URL: <https://github.com/watson-developer-cloud>にて公開

インストール方法

コマンドラインにて
pip install --upgrade watson-developer-cloud

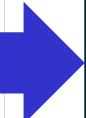
```
1 import json
2 import watson_developer_cloud
3 conversation = watson_developer_cloud.ConversationV1(
4     username='*****',
5     password='*****',
6     version='2017-05-26'
7 )
8 response = conversation.message(
9     workspace_id='*****',
10    input={
11        'text': 'Hello'
12    }
13 )
14 print(json.dumps(response, indent=2))
```

Choregrapheを介さずに用いる場合の
スクリプト

watson公式サイトの
“API リファレンス”に準拠して記述

Usernameやpasswordの取得方法は
この後紹介

Watson Visual Recognitionについて



pepper.png

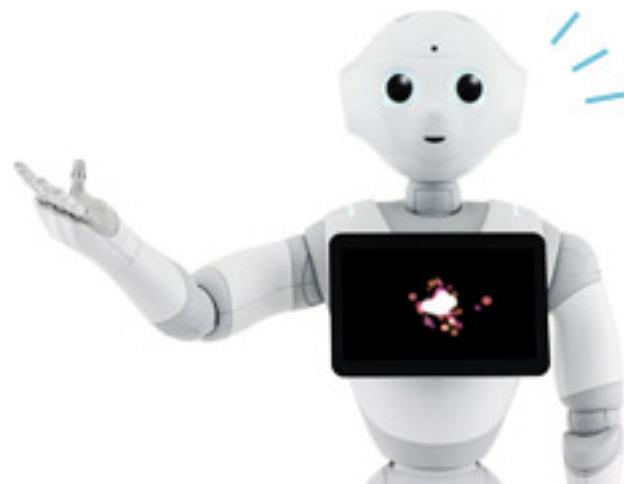
```
{"custom_classes": 0,
"images":
[{"image": "./pepper.png",
"classifiers":
[{"classifier_id": "default",
"name": "default",
"classes":
[{"class": "android", "score": 0.945},
 {"class": "robot", "score": 0.546},
 {"class": "plush (fabric for toys)", "score": 0.5,
 | "type_hierarchy": "/fabric/plush (fabric for toys)" },
 {"class": "fabric", "score": 0.5},
 {"class": "mechanism", "score": 0.801},
 {"class": "alabaster color", "score": 0.918},
 {"class": "ivory color", "score": 0.847}]]}],
"images_processed": 1}
```

送られてきた画像の情報を分類し、

該当しそうなラベルを"class"という形で列挙して返す。

各classの"score"(0~1の値)は、その画像とclassの結びつきの確証度を表す。

**Watsonにログイン
してみよう**



My IBMにログイン

ようこそ [REDACTED] 様

IBMid: [REDACTED]

プロファイル 製品およびサービス

連絡先 e- メール: [REDACTED].com

まだ製品もサービスも追加されていません。

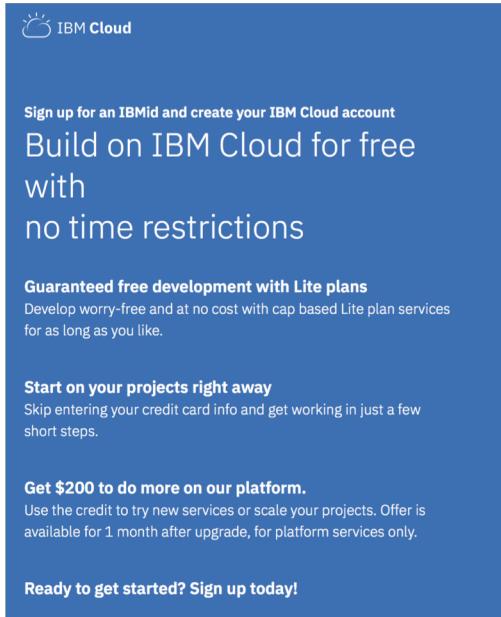
IBM マーケットプレイスの探索

クリック !

そのあとは「IBM Cloud」>「無料評価版を始める」を選択

IBM Cloud登録画面

MyIBMのアカウントとIBM cloudのアカウントは、別物



Already have an IBM Cloud account? [Log in](#)

Email* →

First Name*

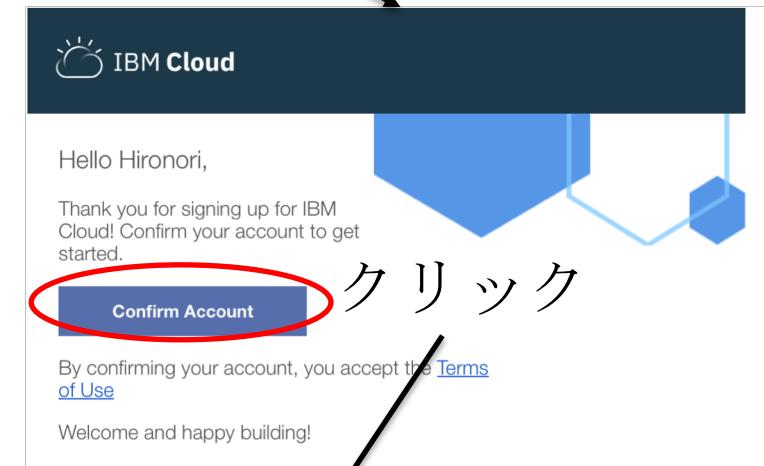
Last Name*

Company

Country or Region*
Japan

Password*

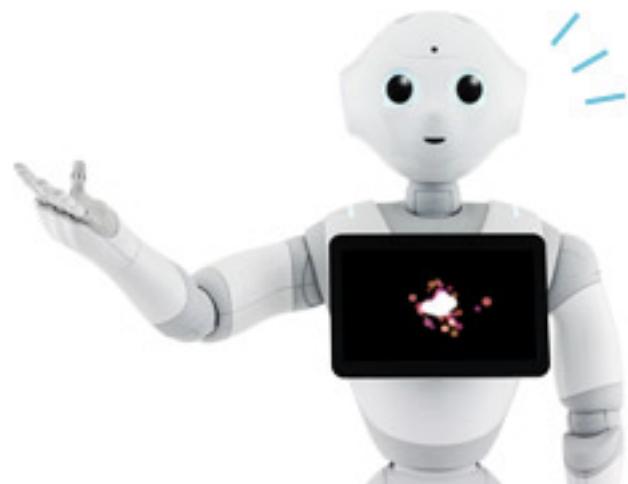
Emailを入力すると



クリック

MyIBMへのログインを要求される

**Watson Visual
Recognitionを
アクティブにしよう**



ダッシュボード

ダッシュボード=現在アクティブになっているサービスの一覧画面

The screenshot shows two views of the IBM Cloud dashboard. The left view is the main dashboard with sections for 'Fast-track your app development' (including 'Build with Watson' and 'Create enterprise-level web apps'), and a central area for 'Service Deployment' with a green circle highlighting the 'Resource Creation' button. The right view is a catalog search interface where a red box highlights the search bar with the query 'label:ライト visual'. A red box also highlights the 'Visual Recognition' service entry in the results list, which is described as a service for detecting faces, objects, and text in images.

console.bluemix.net

My IBM My IBM ダッシュボード - IBM Cloud LinkedIn LinkedInにサインイン Chiyoda Ku ZIP Code - Japan +

IBM Cloud カタログ 資料 サポート 管理 リソースの検索... Suica bar's Account

ダッシュボード

リソース・グループ CLOUD FOUNDRY 組織 CLOUD FOUNDRY スペース 場所 カテゴリー

すべてのリソース すべての組織 すべてのスペース すべての場所 すべてのカテゴリー

サービス立ち上げ リソースの作成

Fast-track your app development

Want to go from prototype to production with the click of a button? Hit the ground running with one of our IBM Cloud starter kits. You can also check out our solution tutorials that provide real-world cloud solution examples.

Build with Watson

Starter Kit - Lite Services · IBM Accelerate your AI development to build smarter solutions.

View Watson starter kits

Create enterprise-level web apps

Starter Kit - Lite Services · IBM Develop and deploy cloud-native apps within minutes.

View App Service starter kits

サービス立ち上げ リソースの作成

console.bluemix.net

My IBM My IBM カタログ - IBM Cloud LinkedIn LinkedInにサインイン Chiyoda Ku ZIP Code - Japan +

IBM Cloud カタログ 資料 サポート 管理 リソースの検索... Suica bar's Account

ライト・プランにより時間制限なく無料で、カタログの最良の機能をお試しください。

ライト・フィルターが有効です。全カタログを表示するには、フィルターを解除してください。

カタログ

label:ライト visual 検索する フィルター

すべてのカテゴリー (1) AI

コンピュート コンピューター ネットワーキング ストレージ AI (1) 分析 データベース Developer Tools 統合 IoT セキュリティーおよび ID スターター・キット Web とモバイル Web とアプリケーション

Visual Recognition

画像コンテンツに含まれる意味を検出します。場面、対象物、顔のイメージ、その他のコンテンツを分析します。既製のデフォルト・モデルを選択するか、独自のカスタム種別を作成します。イメージやビデオ・フ...

さらに詳しい情報

IBM Cloud 試験サービスを確認して、試験ランタイムと試験サービスを試してみましょう。

IBM Cloud 試験サービス

Visual Recognition

The screenshot shows the 'Visual Recognition' service configuration interface. At the top left is the service icon and name. Below it is a large text area describing the service's function: "画像コンテンツに含まれる意味を検出します。場面、対象物、顔のイメージ、その他のコンテンツを分析します。既製のデフォルト・モデルを選択するか、独自のカスタム種別を作成します。イメージやビデオ・フレームの画像コンテンツを解析し、何が起こっている場面なのかを理解するためのスマート・アプリケーションを作成します。". Two specific fields are highlighted with green circles and arrows pointing to them from the Japanese text below: "サービス名: Visual Recognition-it" and "デプロイする地域/ロケーションの選択: ダラス". A 'Default' button is also visible. A 'FEEDBACK' button is at the bottom right.

自由に変更可能（半角英数字）

下にスクロール

ダラスを推奨
(地域によっては
機能が制限される)

「ライト」の選択を確認したら
右下の「作成」をクリック

This screenshot shows the 'Lite' plan details. It includes a checkmark icon, the plan name 'ライト', a description of usage limits ('一般モデル、顔モデルの場合、7,500 個のイベント (イメージ) 1 カスタム・モデル カスタム・モデルの場合、5,000 個のイベント (イメージ) 1 つの Bluemix 組織につき 1 つのライト・プラン・インスタンス'), and a '無料' (Free) status. Below this is a note about API usage and a statement that the 'Lite' plan will be deleted after 30 days if not active.

ライト
一般モデル、顔モデルの場合、7,500 個のイベント (イメージ)
1 カスタム・モデル
カスタム・モデルの場合、5,000 個のイベント (イメージ)
1 つの Bluemix 組織につき 1 つのライト・プラン・インスタンス
無料

ライト・プランでは、ユーザーは 7,500 API呼び出しを無料で(1日あたり 250 呼び出し(分類と Core ML へのエクスポートを含む)まで)実行でき、最大 5,000 イメージを使用してカスタム・モデルのトレーニングを実行できます。ライト・プラン・サービスは、30 日後に削除されます。引き続きサービスを使用するお客様は、標準プランにアップグレードして、新しい API キーを受信する必要があります。

ライト・プラン・サービスは、非アクティブで 30 日経過すると削除されます。

Visual Recognition



Visual Recognition : Visual Recognition-jl

Associated Project : none

Overview

Credentials

ここから各種分類器を
テストすることができる

Custom

Create custom, unique visual classifiers. Use the service to recognize custom visual concepts that are not available with general model.

Create Model +

分類器を自分で
作る

General

[Copy model ID](#)

Generate class keywords that describe the image. Use your own images, or extract relevant image URLs from publicly accessible webpages for analysis.

Test

画像の内容を一般的な
「モノ」から大まかに
判定するモデル

Food BETA

[Copy model ID](#)

Utilize a specialized vocabulary of over 2000 foods to identify meals, food items, and dishes with enhanced accuracy.

Test

食べ物に特化した
学習済みモデル

ここで少し寄り道

General

画像を ドラッグ & ドロップ

Overview Test Implementation

Filter

Threshold 0
0 1

Class

Drop images here to let the classifier analyze them or [browse](#) to select files.

Overview Test Implementation

[Code Snippets](#)

pip

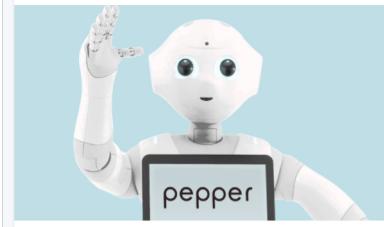
```
pip install --upgrade "watson-developer-cloud>=1.0,<2.0"
```

Authentication

```
from watson_developer_cloud import VisualRecognitionV3
visual_recognition = VisualRecognitionV3(
    '2016-05-20',
    api_key='{api-key}'
)
```

Clear results

img.jpg



pepper

ivory color	0.97
controller	0.64
robot	0.63
timer	0.63
mechanism	0.60
tool	0.58
android	0.56

Watson developer cloud(WDC)の
簡易APIリファレンスが
参照できるので、
WDCを用いる場合は
この記述に従えばよい
ただしAPI KEYが必要

API keyの取得

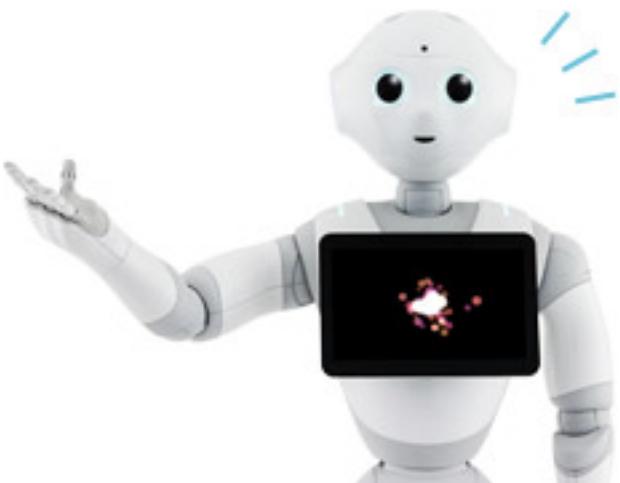
“General”から離れ、visual recognitionのホーム画面へ

The screenshot shows the 'Service Credentials' section of the Watson Visual Recognition service. At the top, there are tabs for 'Overview' and 'Credentials', with 'Credentials' being the active tab and circled in red. Below the tabs is a search bar labeled 'Search'. A table lists credentials with columns for 'Key name' (apsx-data) and 'Date created' (2018-03-25T08:19:23Z). Under the 'Actions' column for the first row, there is a link 'View credentials' which is also circled in red. To the right of the table, there is a JSON representation of the credential data:

```
{  
  "url": "https://gateway-a.watsonplatform.net/visual-recognition/api",  
  "note": "It may take up to 5 minutes for this key to become active",  
  "api_key": "目当てのAPI-key"  
}
```

Rest APIを用いる
場合のターゲット
URL

Node-redからサービスを利用する場合は必ずしも必要でない
(後ほど紐づける方法を案内)

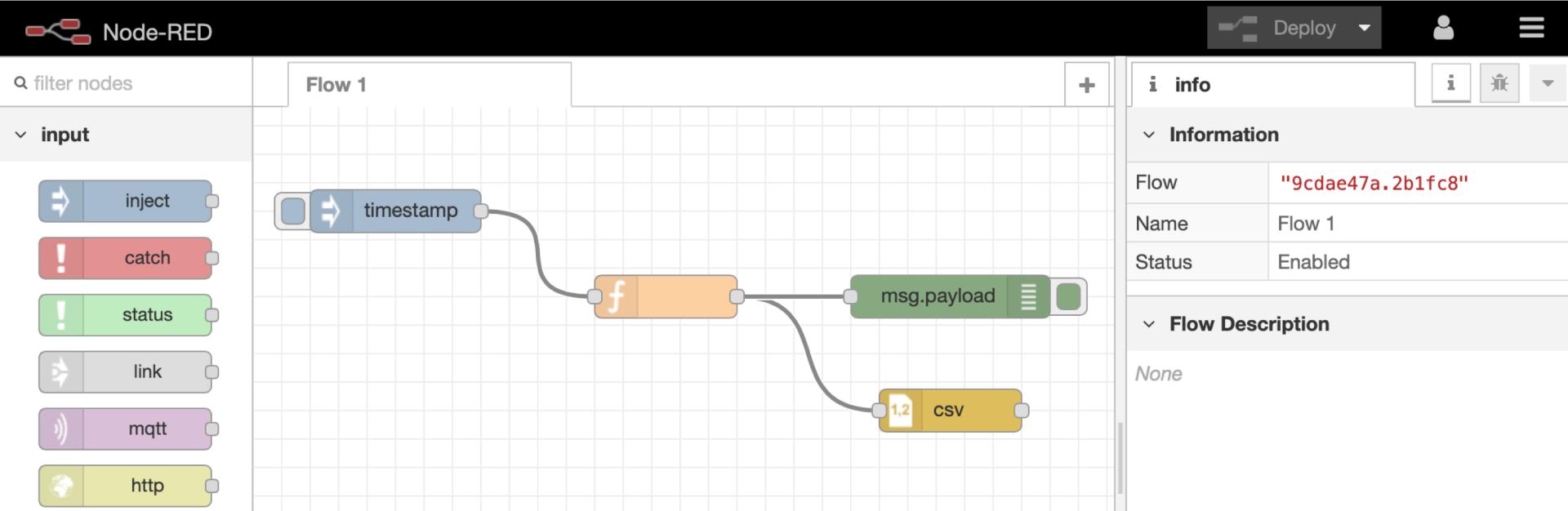


Node-redを
アクティブにしよう

Node-redとは

IBMで開発されたオープンソースのGUIの開発環境。
IoT分野で好んで用いられる。

Choregrapheにおけるボックスに相当する「ノード」 同士を線でつなぐことでプログラムを作る。



Node-redを立ち上げる

The screenshot shows the IBM Cloud Dashboard. At the top, there are filters for REGION (US South), CLOUD FOUNDRY ORG (@gmail.com), and CLOUD FOUNDRY SPACE (dev). A green oval highlights the "Create resource" button. A large black arrow points from this button down to the "Node-RED Starter" service card in the catalog below.

Dashboard

RESOURCE GROUP: All Resources | REGION: US South | CLOUD FOUNDRY ORG: @gmail.com | CLOUD FOUNDRY SPACE: dev | Create resource

サービス 2/100 使用済み

名前	サービス・オファリング	プラン	操作
DSX	Watson Studio	Lite	⋮
Visual Recognition-jl	Visual Recognition	Lite	⋮

ボイラープレート
今すぐ新しいアプリの作成を始めましょう。

 Internet of Things Platform Starter Bluemix で Node-RED を使用して Internet of Things Platform アプリケーションを作成する ライト IBM	 Node.js Cloudant DB Web Starter Cloudant NoSQL DB サービスを 'SDK for Node.js™' ランタイムと一緒に使用する ライト IBM	 Node-RED Starter This application demonstrates how to run the Node-RED open-source application ライト コミュニティー
 Python Flask A simple Python Flask application that will get you up and running ライト コミュニティー	 Ruby Sinatra Sinatra フレームワークを使用して Ruby Web アプリケーションを開発する ライト コミュニティー	

Node-redの立ち上げ

IBM Cloud カタログ 資料 サポート 管理 リソースの検索... Suica bar's Account

すべて表示

Cloud Foundry アプリの作成

Node-RED Starter

This application demonstrates how to run Node-RED open-source project within IBM Cloud.

ホスト名: pepper-visual
ドメイン: mybluemix.net

資料の表示

バージョン: 0.8.1
タイプ: ポイント-to-Point
場所: ダラス, シンシナティ, フランクフルト, ロンドン, ワシントン DC, ダラス

デプロイする地域/ロケーションの選択: [Redacted]
スペースの選択: dev

選択済みプラン:

SDK for Node.js™: ライト
Cloudant: Lite

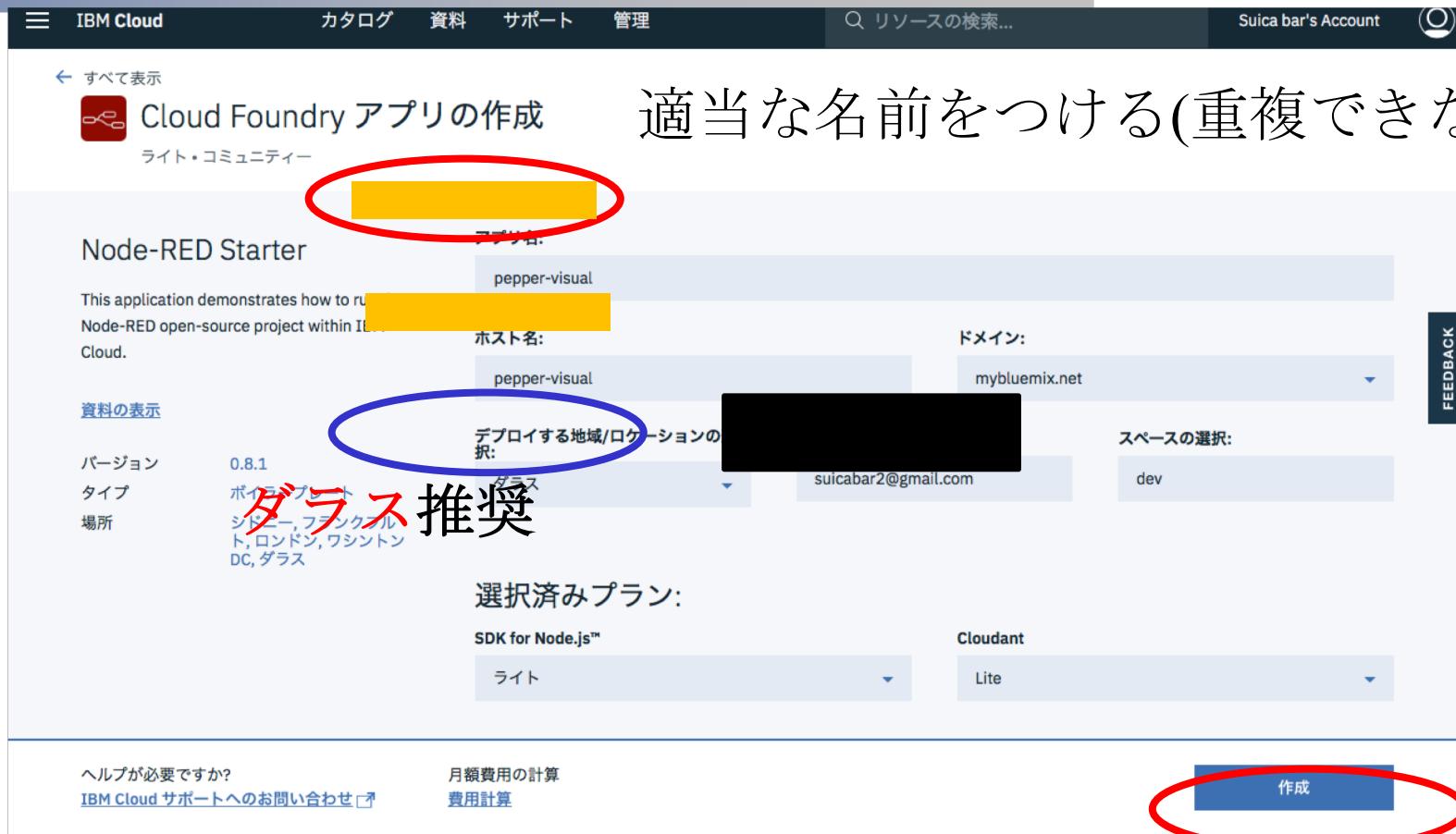
ヘルプが必要ですか?
[IBM Cloud サポートへのお問い合わせ](#)

月額費用の計算
[費用計算](#)

作成

適当な名前をつける(重複できない)

押したらしばらく待つ



Node-redとVisual Recognitionサービスの紐づけ

The screenshot shows the IBM Cloud Platform interface for a Cloud Foundry application. On the left, there's a sidebar with navigation links: 開始, 概要, ランタイム, and 接続 (which is highlighted with a red box). Below the sidebar, there's a section for '接続' (Connections) with a heading '① 「接続」をクリック' (Click 'Connections'). This section displays the following details:

- ビルドパック: SDK for Node.js™
- インスタンス: 1 (The first instance has a green circle around it)
- MB (インスタンス当たりのメモリー): 256
- MB (合計の割り振り): 256 (This value is also circled in green)

At the top of the main dashboard, there are tabs for カタログ, 資料, サポート, and 管理.

② 「接続の作成」をクリック

A modal dialog box titled '接続の作成' (Create Connection) is shown. It contains a search bar labeled '項目のフィルター' and a large blue button labeled '接続の作成 +'. The entire '接続の作成' button is highlighted with a red box.

③先ほど立ち上げた
Visual Recognition
のサービスを選択

この操作によってノードにAPI keyを入力する必要がなくなる

Node-redの立ち上げ

IBM Cloud

カタログ 資料 サポート 管理

Cloud Foundry アプリ /

開始 概要 ランタイム 接続 ログ モニタリング API Management

経路

組織: 場所: US South スペース: dev

クリック>~~~.mybluemix.net/をクリック

ランタイム

ビルダック
SDK for Node.js™

インスタンス
すべてのインスタンスが実行中です
正常性は 100% です

MB (インスタンス当たりのメモリー)
MB (合計の割り振り)
0 MB (使用可能)

The screenshot shows the IBM Cloud dashboard for a Cloud Foundry application named 'Route'. On the left, there's a sidebar with links for '開始', '概要', 'ランタイム', '接続', 'ログ', 'モニタリング', and 'API Management'. The '概要' tab is selected. In the center, there's a summary card with a yellow header bar. Below it, there's a dropdown menu labeled '経路' with a red box around it, followed by a blue button with three dots. To the right of this is a status message: 'アプリは稼働中ですが、開発が非アクティブになってます' (The app is running but development is inactive). Below the dropdown are fields for '組織:' (Organization), '場所: US South' (Location), and 'スペース: dev' (Space). A large red text overlay on the right side of the card says 'クリック>~~~.mybluemix.net/をクリック' (Click >~~~.mybluemix.net/ to click). At the bottom, there's a section titled 'ランタイム' (Runtime) with four circular icons: a blue one for 'ビルダック' (Builder) with 'SDK for Node.js™', a white one for 'インスタンス' (Instance) with '1' and a plus/minus sign, a blue one for 'MB (インスタンス当たりのメモリー)' (Memory per instance) with '256', and a green one for 'MB (合計の割り振り)' (Total allocation) with '256'.

Node-redの立ち上げ

Welcome to your new Node-RED instance on IBM Cloud

We know you're eager to start wiring up your flows, but first there are a couple of tasks you should do:

- Secure your Node-RED editor
- Browse available IBM Cloud nodes

Progress bar: ● ○ ○ ○

Previous Next



Secure your Node-RED editor

- Secure your editor so only authorised users can access it

Username

Password

Must be at least 8 characters

Allow anyone to view the editor, but not make any changes

- Not recommended: Allow anyone to access the editor and make changes

Progress bar: ● → ○ ○ ○

Previous Next



usernameとPasswordは自身で設定する

Node-redの立ち上げ

Browse available IBM Cloud nodes

There are lots of nodes available from the community that can be used to add more capabilities to your application. The list below is just a small selection.

You can find many more nodes on the [Flow Library](#).

You can use the Palette Manager built into editor to search for and install nodes. Alternatively, you can also edit your application's package.json file and adding them to the dependencies section.

[node-red-dashboard](#)

Quickly create dashboards driven by Node-RED

[node-red-contrib-ibm-wiotp-device-ops](#)

Perform device and gateway operations using the Watson IoT Platform

[node-red-contrib-iot-virtual-device](#)

Simulate device behavior and use it to run many device instances

[node-red-contrib-objectstore](#)

Store, delete and restore objects in the ObjectStore service

[node-red-contrib-bluemix-hdfs](#)

Create and update files using HDFS for Analytics

[node-red-contrib-ibmpush](#)

Send push notifications to mobile devices using the

Previous

Next

Finish the install

You have made the following selections:

- Secure your editor so only authorised users can access it

You can change these settings at any time by setting the following environment variables via the IBM Cloud console:

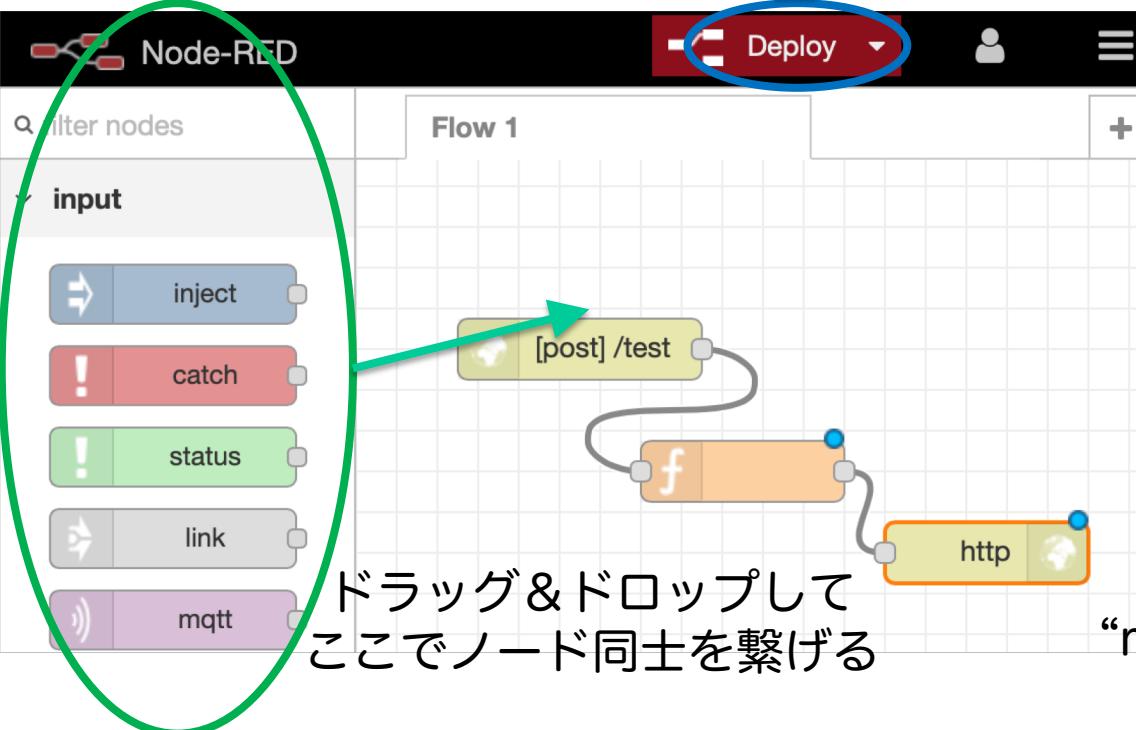
- NODE_RED_USERNAME - the username
- NODE_RED_PASSWORD - the password
- NODE_RED_GUEST_ACCESS - if set to 'true', allows anyone read-only access to the editor

Previous

Finish

Node-redの基礎知識

ノードを繋いだのち
ここをクリックで更新



線で繋がれたノード同士では
以下のような
Json形式で情報のやりとりを行う

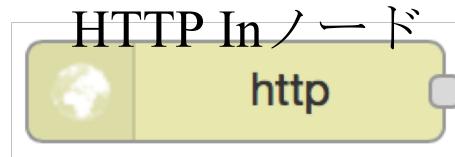
msgオブジェクト

msg.payload

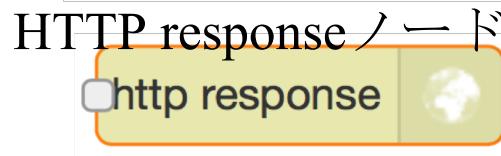
msg.XXXX

“msg”はやり取りするjson全体の名前
その中の”payload”はメールに
おける「本文」に相当

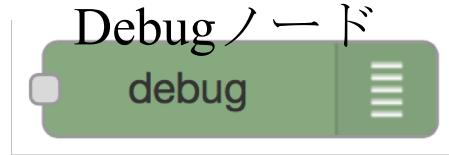
今回使用するノード



httpエンドポイントを作り、情報を受け取る



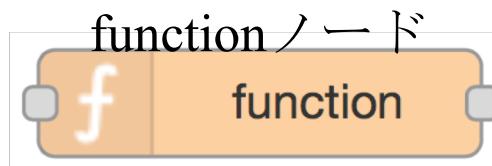
HTTP In ノードで受けた情報の発信元に出力を返す



画面右側の「デバッグ」タブに
msg.payloadの内容を表示

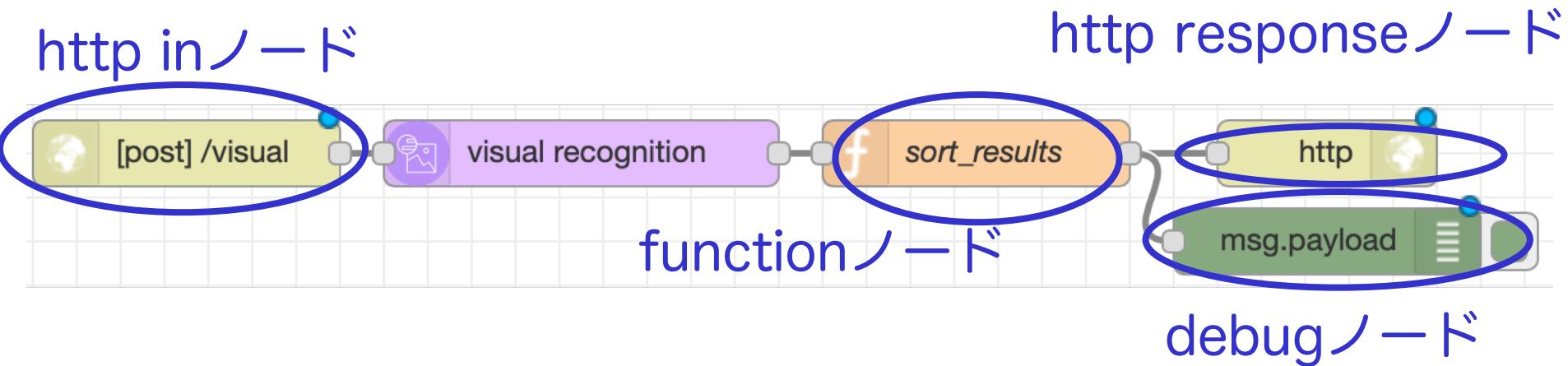


画像を入力するとWatson Visual Recognitionの機能を
用いて判別した結果が出力される



javascriptで中の処理を記述する

今回の画像判別処理の組み方



http inノードの設定

Edit http in node

Delete Cancel Done

▼ node properties

Method POST

Accept file uploads?

URL visual

Name Name

Visual Recognitionノードの設定

visual recognition ノードを編集

削除

中止

完了

▼ プロパティ

Detect:

Classify an image

Name

Name

Language

English

usernameとAPI keyを
要求される場合は、
Visual RecognitionとNode-redの
接続がうまくいっていない
node-redを再始動するか
API keyを直接入力することが必要

Watson Visual Recognitionについて（おさらい）



pepper.png

```
{"custom_classes": 0,  
"images":  
[{"image": "./pepper.png",  
"classifiers":  
[{"classifier_id": "default",  
"name": "default",  
"classes":  
[{"class": "android", "score": 0.945},  
 {"class": "robot", "score": 0.546},  
 {"class": "plush (fabric for toys)", "score": 0.5, "type_hierarchy": "/fabric/plush (fabric for toys)"},  
 {"class": "fabric", "score": 0.5},  
 {"class": "mechanism", "score": 0.801},  
 {"class": "alabaster color", "score": 0.918},  
 {"class": "ivory color", "score": 0.847}]]}],  
"images_processed": 1}
```

このjsonファイルから、欲しい情報を抽出する必要がある。
これを次ページのfunctionノードで記述



functionノードの記述

↓配布資料「watson_visual.txt」からコピペ↓

```
data=msg.result["images"][0]["classifiers"][0]["classes"]
```

```
data.sort(function(val1,val2){  
    val1 = val1.score;  
    val2 = val2.score;  
    if( val1 < val2 ) {  
        return 1;  
    }  
    else {return -1; } });
```

visual recognitionノードの出力は元々
確証度の高い順（降順）になっていない
ため、ここで降順にソートしている

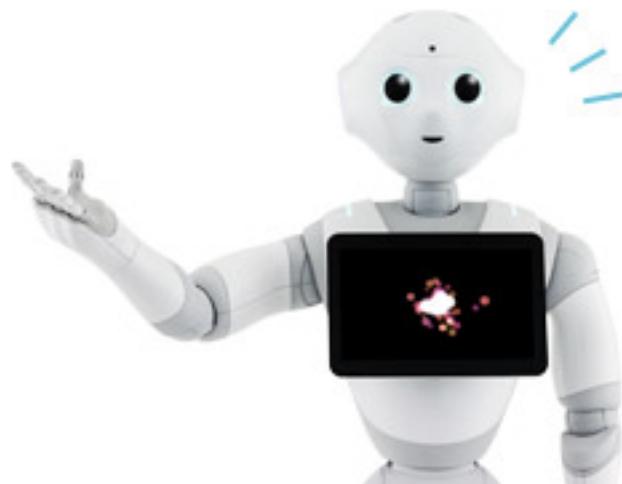
```
var list = data[0]["class"]+" or "+data[1]["class"]+" or "+data[2]["class"];  
msg.payload=list;  
return msg;
```

3大候補をorでつないだ文字列を出力



ノードを編集後は「デプロイ」を忘れない！

Choregraphe側の 設定



HTTP通信を行うためのボックスを外部からインポートしよう

<https://github.com/takujikawata/webapiboxlibrary>

The screenshot shows the GitHub repository page for 'takujikawata / webapiboxlibrary'. The repository has 5 commits, 1 branch, 0 releases, and 2 contributors. The 'Clone or download' button is highlighted with a red box and a blue arrow pointing to it from below. The page also shows a commit history with two entries: 'Update README.md' by takujikawata and 'WebSocket ボックスの NAOqi 2.5.5 対応。NAOqi2.5.5 には古いバージョンの six.py が入っているので...' by takujikawata.

takujikawata / webapiboxlibrary

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Web API を使うような Pepper アプリを作りたい時に役に立つ Box ライブライ

5 commits 1 branch 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

takujikawata Update README.md

README.md Update README.md a year ago

webapibox.cbl WebSocket ボックスの NAOqi 2.5.5 対応。NAOqi2.5.5 には古いバージョンの six.py が入っているので... a year ago

Latest commit fb09d90 on 30 Apr 2017

ここからダウンロードして保存

HTTP通信を行うためのボックスを外部からインポートしよう

<https://github.com/takujikawata/webapiboxlibrary>



takujikawata / **webapiboxlibrary**

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

Web API を使うような Pepper アプリを作りたい時に役に立つ Box ライブリ

ここからダウンロードして保存

5 commits 1 branch 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

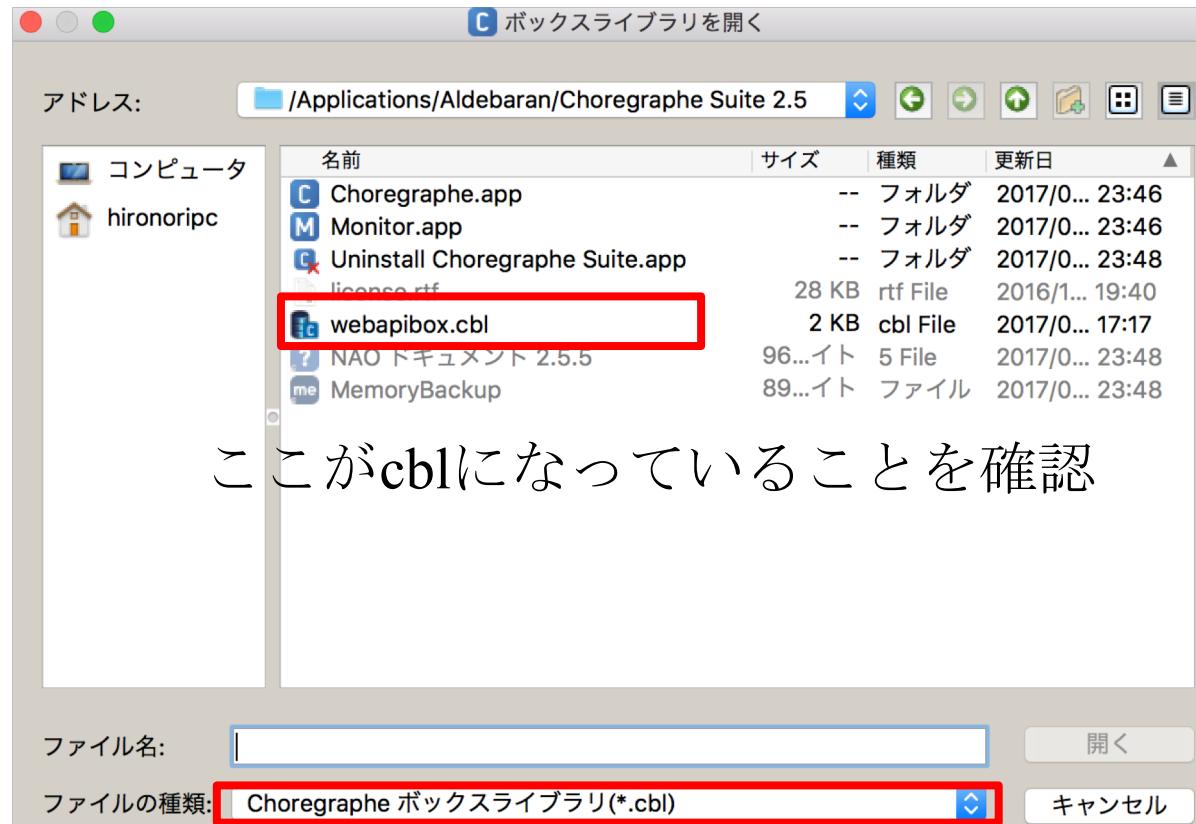
takujikawata Update README.md

Latest commit fb09d90 on 30 Apr 2017

README.md Update README.md a year ago

webapibox.cbl WebSocket ボックスの NAOqi 2.5.5 対応。NAOqi2.5.5 には古いバージョンの six.py が入っているので... a year ago

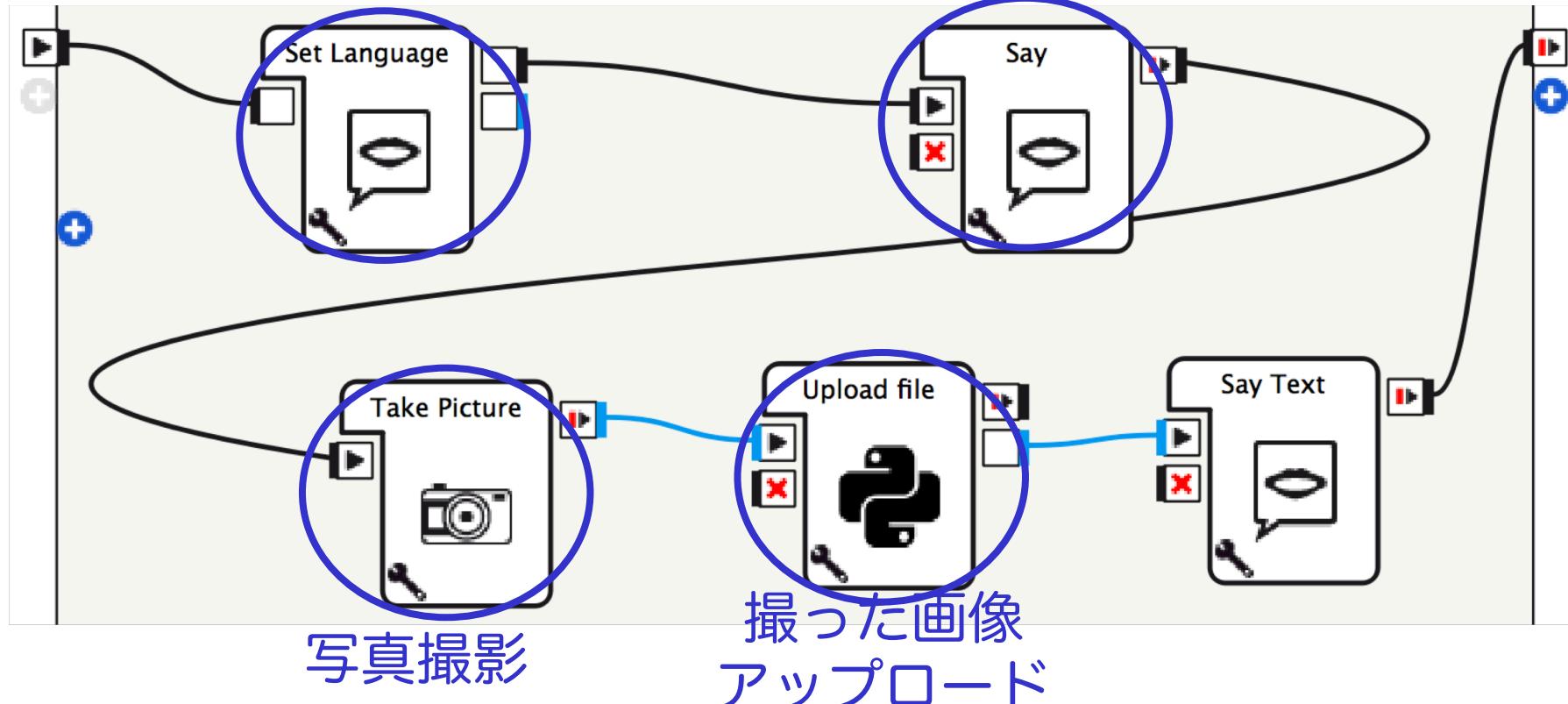
cblファイルのインポート



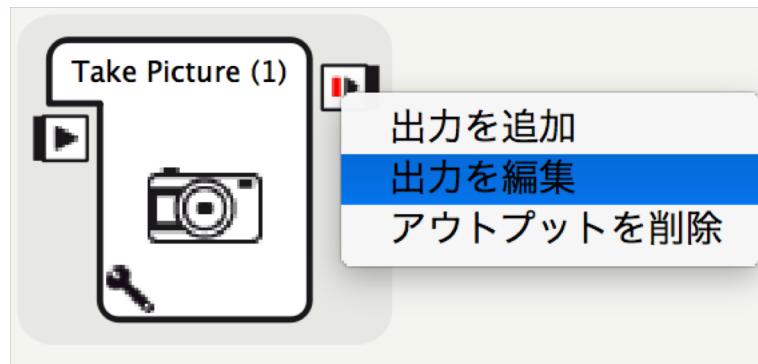
写真を撮ってNode-redにPostでアップロード

英語に設定変更

"Say Cheese!"



Take pictureボックスの編集



出力端子を編集し、
タイプを「文字列」に変更する

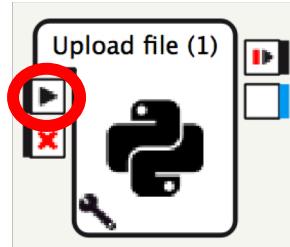
self.onStopped()



43行目に引数を追加

self.onStopped(self.recordFolder+fileName+".jpg")

Upload Fileの設定



入力のタイプを「文字列」に変更

upload url

src file

content type

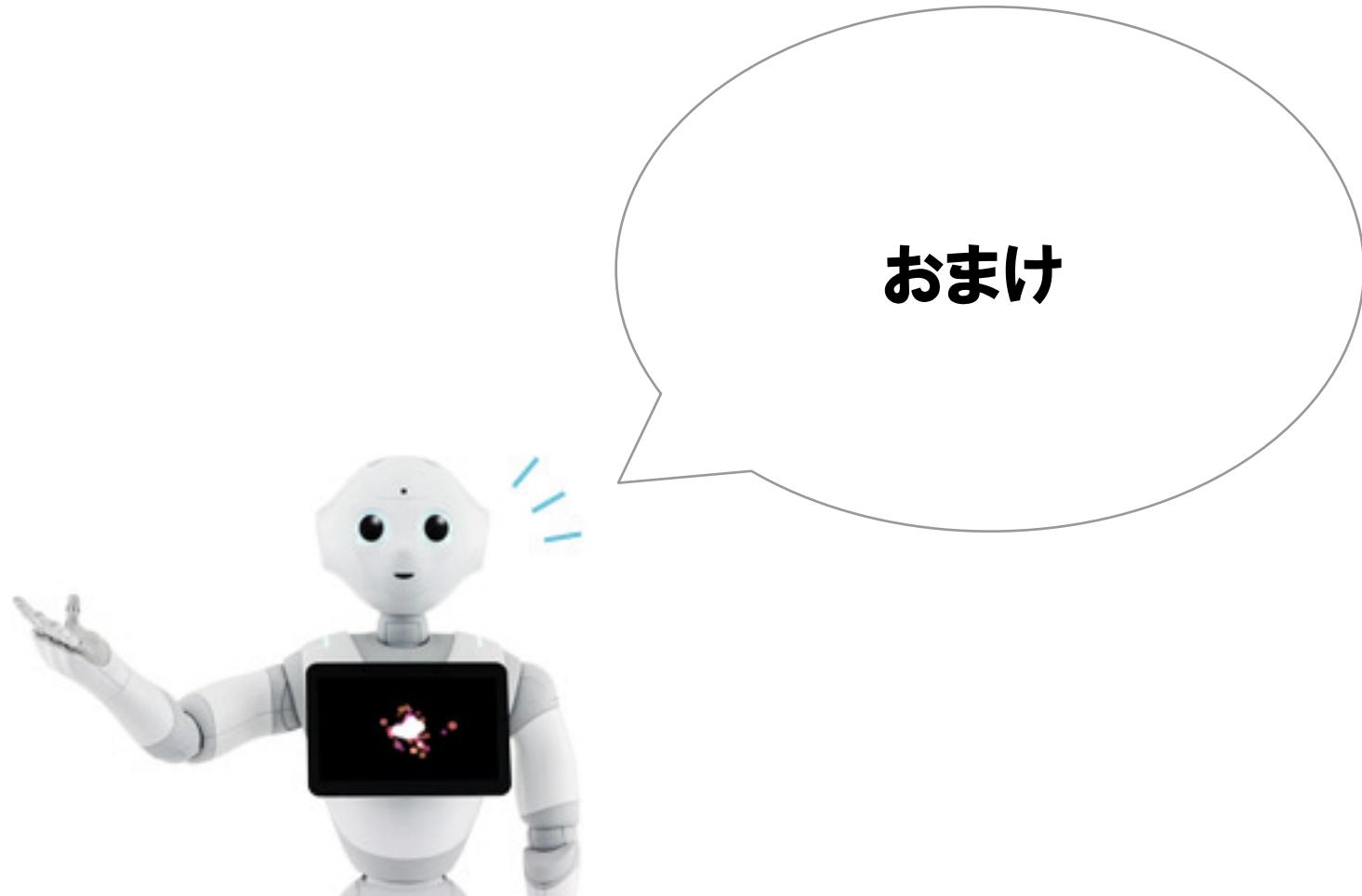
ロボット上の変数の自動更新

upload urlは
https://自身のnoderedのurl/Post名

content typeは
image/jpg

18行目 `def onInput_onStart(self, p):` 引数pを追加

22行目 `srcFile = p` srcFileに引数pを代入



Pepper デベロッパー ポータル

「Pepper developer」で検索
<https://developer.softbankrobotics.com/jp-ja>

Pepperに関するデベロッパー向けの情報を集約したポータルサイト

- ・技術ドキュメント
- ・事例を共有するショーケース
- ・Pepper SDK for Android Studioのダウンロード
- ・最新ニュースの提供

Pepper アトリエ秋葉原 with SoftBank

「アトリエ秋葉原 ブログ」で検索

- ・ペッパー開発に役立つ記事を見ることができる
- ・イベントの紹介とイベントのレポートが見ることができる
- ・tipsの項目から開発に便利なツールを手に入れることができる

アトリエ秋葉原FBグループ

「アトリエ秋葉原 FB」で検索

- ・アトリエ秋葉原のFacebookグループです
- ・情報共有や質問ができます

Qiita

「Qiita pepper」で検索

- ・エンジニアの情報交換サイト
- ・PepperタグでPepperに関する様々な技術情報がある