

Bluemix Hands-On #2

Kifumi Numata
University Relations, IBM Japan



本日のハンズオン

- Bluemix へのログイン
- Node-REDでHello World!
- IBM Watsonとは
- Watson Visual Recognition APIを使った画像認識アプリの作成

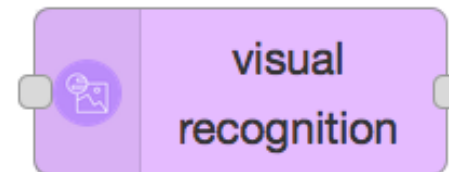
Watson Visual Recognition APIをBluemixのNode-REDから呼び出そう!

概要

- IBM Watsonのサービスの一つであるVisual Recognition APIを呼び出す簡単なサンプルです。
- Node-REDを使って簡単に呼び出しているのが特徴です。IBM Bluemixを使えば簡単に、迅速にアプリケーションを作ることが可能です。

ノード解説

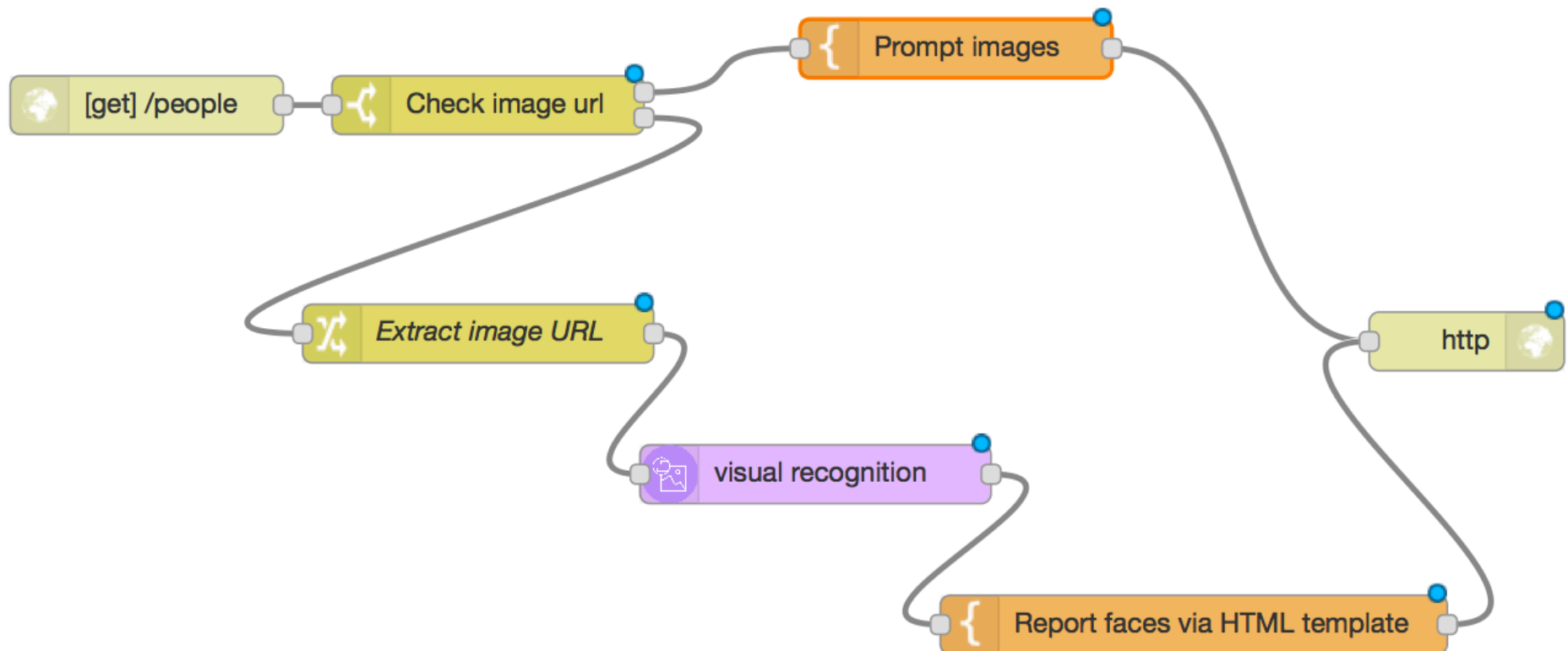
Visual recognition node



IBMがBluemixは様々なコグニティブAPIを提供しています。その中でも画像認識サービスである、Visual Recognition は画像解析から年齢や人物判定まで行う機能を持ったサービスです。IBM Watsonのカテゴリに入っているので確認してみてください。

全体フロー概要

- 画像のURL（例："http://xxxxx.jpg"）をVisual Recognition のAPIにかけると画像解析を行い、顔認識の結果を返してくれるサンプルアプリです。



1. BluemixでNode-REDサービスを設定する

- Bluemix Hands-On #1 の資料を参照してください。

2. Visual Recognition APIを追加する

- Node-REDのノードに画像認識のための Visual Recognition があるのですが、このままでは使えません。このNode-REDのアプリケーションにVisual Recognition APIを追加してあげる必要があります。
- Bluemixのメニュー画面左上の「IBM Bluemix」をクリックし、「すべてのアプリ」一覧のなかから、先ほどのNode-REDのアプリケーションをクリックしてください。

The screenshot shows the IBM Bluemix console interface. At the top, there's a dark navigation bar with the 'IBM Bluemix' logo and text highlighted by a red box. Below this is a search bar and a 'すべてのアプリ (11)' section. A table titled 'Cloud Foundry アプリ 5.500 GB/8 GB 使用済み' displays a list of applications. The first row, 'AlchemyAPIkn', is highlighted with a red box. The table columns are: 名前, 経路, メモリー (MB), インスタンス, 実行中, 状態, and アクション.

名前	経路	メモリー (MB)	インスタンス	実行中	状態	アクション
AlchemyAPIkn	AlchemyAPIkn.mybluemix.net	512	1	1	● 実行中	🔄 📄 ⋮
BB8-no3	BB8-no3.mybluemix.net	512	1	1	● 実行中	🔄 📄 ⋮

2. Visual Recognition APIを追加する

- 左側の「接続」をクリックし、右側の「新規に接続」のアイコンをクリックします。



2. Visual Recognition APIを追加する

- 左側の「Watson」をクリックし、「Visual Recognition」を選択します。

The screenshot displays the IBM Watson Developer Cloud interface. On the left, a sidebar lists various categories under 'すべてのカテゴリー' (All Categories). The 'Watson' category is highlighted with a red box. The main area shows a grid of Watson services. The 'Visual Recognition' service is highlighted with a red box. Each service card includes an icon, a title, a brief description, and the IBM logo.

すべてのカテゴリー

インフラストラクチャー

計算

ストレージ

ネットワーク

セキュリティ

アプリ

ポイラプレート

Cloud Foundry アプリ

コンテナ

OpenWhisk

モバイル

サービス

データ & 分析

Watson

すべてのインターネット

API

ネットワーク

AlchemyAPI

構造化されていないテキストおよびイメージのコンテンツを分析する AlchemyAPI サービス

IBM

Conversation

アプリケーションに自然言語インターフェースを追加してエンドユーザーとのやり取り

IBM

Document Conversion

HTML、PDF、Microsoft Word™ 文書を正規化された HTML、プレーン・テキスト、JSON

IBM

Language Translator

特定のドメインのテキストをある言語から他の言語に翻訳します。

IBM

Natural Language Classifier

Natural Language Classifier は、質問テキストに対して自然言語の分類を実行します。

IBM

Personality Insights

Watson Personality Insights は、トランザクション・データやソーシャル・メディアの

IBM

Retrieve and Rank

機械学習の拡張検索機能をアプリケーションに追加します

IBM

Speech To Text

低遅延、ストリーミング・トランスクリプション

IBM

Text to Speech

テキストから自然な音声を合成します。

IBM

Tone Analyzer

Tone Analyzer は、言語分析を行って、コミュニケーションから感情、社交性、文体の

IBM

Tradeoff Analytics

対立する複数の目標がある中で、より良い選択を行えます。トレードオフを検討するた

IBM

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、および

IBM

2. Visual Recognition APIを追加する

- 「作成」をクリックします。

- 「アプリケーションの再ステージ」のポップアップ画面が現れるので「再ステージ」をクリックします。

- 再ステージングし正常に再起動すればOKです！

3. Node-REDでプログラミング

- 「アプリの表示」をクリックし、「Go to your Node-RED flow editor」をクリックして、Node-REDが起動します。



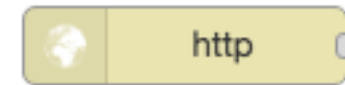
- Node-REDエディターが立ち上がったら、上側右寄りの「+」をクリックして、新しいフロー画面「Flow 2」を立ち上げます。



3-1. HTTP Input node

- Visual Recognition API は REST の GET メソッドでアクセスして画像を解析します。

- 左側のパレットの Input カテゴリ内の http の node をドラッグ&ドロップし、キャンバス内に配置します。



- プロパティ内のURL欄にアクセスポイントを記載します。
ここでは /callwatson とでもしておきます。

- Name の欄はノードの名前をわかりやすいようにしておくために記述しておきます。
任意ですが、ここでは HTTP Input にしておきます。

Edit http in node

CancelDone

Method

GET

URL

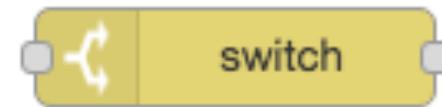
/callwatson

Name

HTTP Input

3-2. switch node

- 画像のURLをチェックするノードを準備します。
- 左側のリソースパレットの function カテゴリ内の switch ノードをフローエディタ中央のキャンバスにドラッグ&ドロップします。
- プロパティ内の左下にある「+add」をクリックして、分岐ロジックを2つ用意します。
- Propertyは、右図の通りに imageUrl属性に含まれるペイロードのnullチェックを行います。
- nullであれば、"1"にそれ以外であれば"2"に値が渡されます。



Edit switch node

Cancel Done

Name Check image url

Property msg. payload.imageUrl

is null	→ 1	x
otherwise	→ 2	x

3-3. template node (初期画面)

- 画面のHTMLを表示したり、Inputとなる画像を送信するためのメニューを提供するためにHTMLを記述します。
- templateノードをフローエディタ中央のキャンバスにドラッグ&ドロップします。



- プロパティを以下のように記述します。

```
<h1>Welcome to a Watson Visual Recognition sample Face Detection app</h1>
<H2>Recognize anyone?</H2>
<form action="{{req._parsedUrl.pathname}}">
  
  
  
  
  <br/>Right-click one of the above images and select Copy image location and paste the URL in the box below. <br/>Do an image
  search for faces, try multiple faces. After you click on an image, to the right it usually says "View image" click that to get the
  URL. <br/>
  <br/>Image URL: <input type="text" name="imageurl"/>
  <input type="submit" value="Analyze"/>
</form>
```

3-3. template node (初期画面)

Edit template node

Cancel

Done

Name

Prompt images

Set property

▼ msg. payload

Template

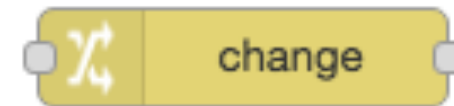
Syntax Highlight: mustache

```
1 <h1>Welcome to a Watson Visual Recognition sample Face Det
2 <H2>Recognize anyone?</H2>
3 <form action="{{req._parsedUrl.pathname}}">
4   Right-click one of the above images and selec
9   <br>Image URL: <input type="text" name="imageurl"/>
10  <input type="submit" value="Analyze"/>
11 </form>
```

3-4. change node

- 入力画面から画像URLを抽出するchangeノードを定義します。

- 左側のリソースパレットのfunction カテゴリ内のchange ノードをフロー・エディタ中央のキャンバスにドラッグ&ドロップします。



- ここからpayload属性をimageurl属性に変換します。
以下の通りにプロパティを設定します。

Edit change node

Cancel Done

Name Extract image url

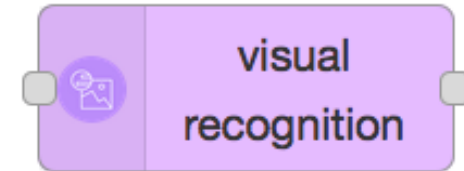
Rules

Set msg. payload

to msg. payload.imageurl

3-5. Image Analysis


- 画像解析のための visual recognition ノードを定義します。
- 左側のリソースパレットの IBM_Watson カテゴリ内の visual recognition ノードをフロー・エディタ中央のキャンバスにドラッグ&ドロップします。
- プロパティでは顔認識を行うため、以下の通りに Detect を Detect Faces に設定します。




Edit visual recognition node

Cancel

Done

 Detect:

Detect Faces

 Name

Name

3-6. template node (結果)

- WatsonのImage Analysisから返ってきた結果を表示させるためのHTMLを記載します。

- templateノードをフローエディタ中央のキャンバスにドラッグ&ドロップします。



- プロパティを以下のように記述します。

```
<h1>Visual Recognition v3 Image Analysis</h1>
  <p>Analyzed image: {{result.images.0.resolved_url}}<br/></p>
  {{^result}}
    <p>No Face detected</p>
  {{/result}}
  <p>Images Processed: {{result.images_processed}}</p>
  <table border='1'>
    <thead><tr><th>Age
Range</th><th>Confidence</th><th>Gender</th><th>Confidence</th><th>Name</th></tr></thead>
    {{#result.images.0.faces}}<tr>
      <td><b>{{age.min}} - {{age.max}}</b></td><td><i>{{age.score}}</i></td>
      <td>{{gender.gender}}</td>
      <td>{{gender.score}}</td>
      <td>{{identity.name}} ({{identity.score}})</td>
    </tr>{{/result.images.0.faces}}
  </table>
  <form action="{{req._parsedUrl.pathname}}">
    <br><input type="submit" value="Try again or go back to the home page"/>
  </form>
```

3-6. template node (結果)

Edit template node

Cancel
Done

Name
Report faces via HTML template

Set property
msg. payload

Template
Syntax Highlight: mustache

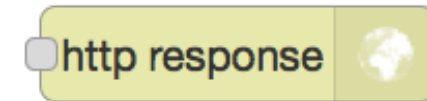
```

1 >Visual Recognition v3 Image Analysis</h1>
2 <p>Analyzed image: {{result.images.0.resolved_url}}<br/><in
3 {{^result}}
4   <p>No Face detected</p>
5 {{/result}}
6 <p>Images Processed: {{result.images_processed}}</p>
7 <table border='1'>
8   <thead><tr><th>Age Range</th><th>Confidence</th><th>Gen
9   {{#result.images.0.faces}}<tr>
10     <td><b>{{age.min}} - {{age.max}}</b></td><td><i>{{
11     <td>{{gender.gender}}</td><td>{{gender.score}}</td>
12     <td>{{identity.name}} ({{identity.score}})</td>
13   </tr>{{/result.images.0.faces}}
14 </table>
15 <form action="{{req._parsedUrl.pathname}}">
16   <br><input type="submit" value="Try again or go back to
17 </form>

```

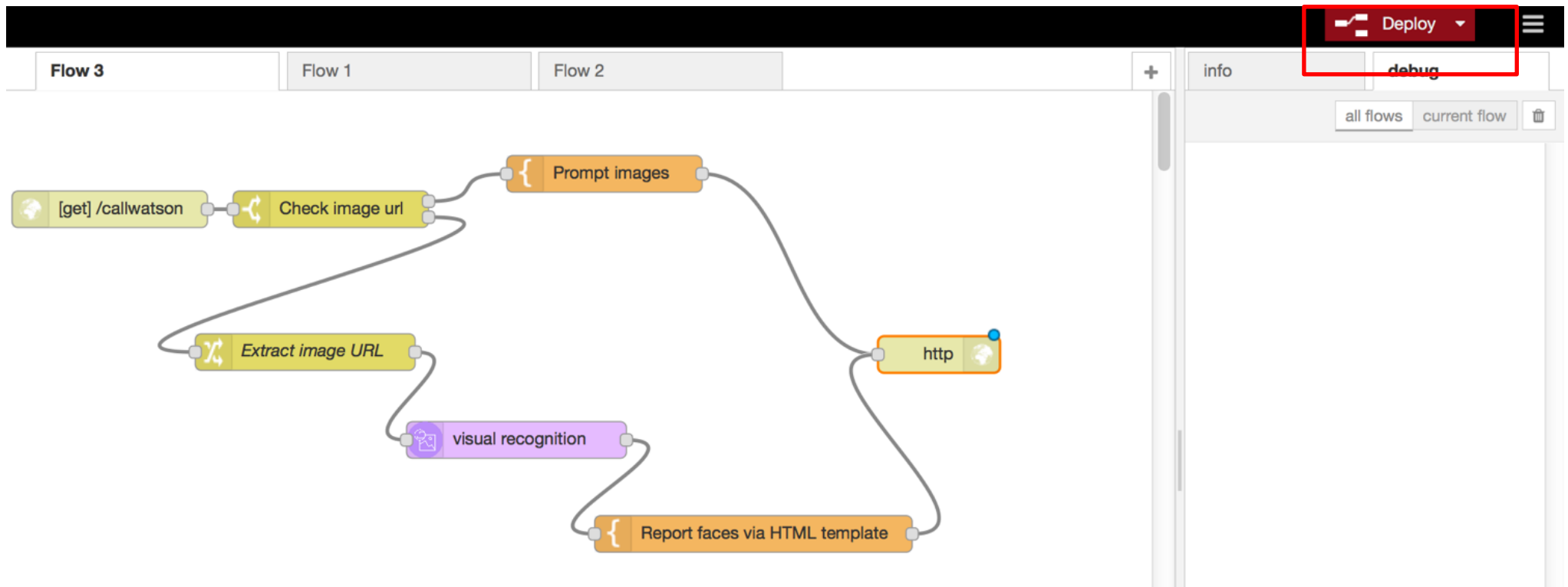
3-7. フローをつなげる

- 最後に http response ノードをフローエディタ中央のキャンバスにドラッグ&ドロップします。



- 出来上がった客ノードをつなげて、右上のDeployをクリックすれば完成です!

エラーが出ていないことを確認してください。



4. 動作確認

- ブラウザのURL欄に <http://xxxx.mybluemix.net/callwatson> をインプットして呼び出してみましょう。
- Image URLの入力欄にWatsonに読ませたい画像URLを入れてみてください。