



マイコンとiPaaSで 呼び出しアプリを作ってみた

~M5StackとAzure Logic Appsを添えて~

@RPACommunity









IT活用ネタ・自動化ネタ
ライトニングトーク
テクノロジーを楽しく学ぶ

RPA COMMUNITY 10/4 20:00~



自己紹介



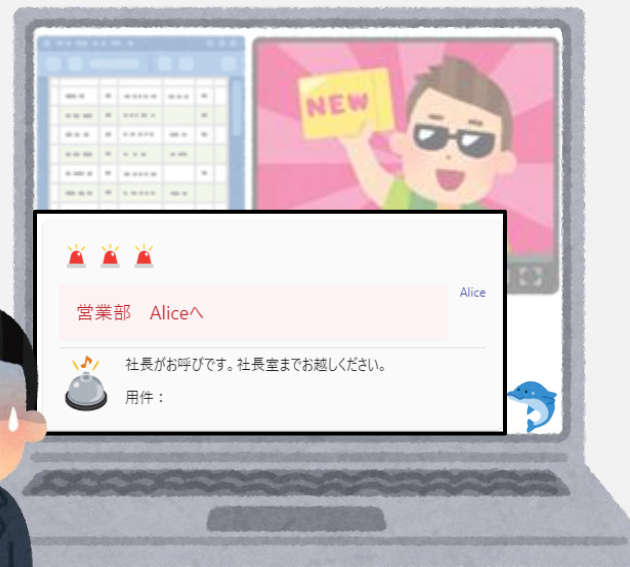
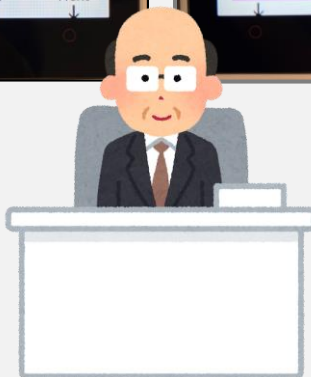
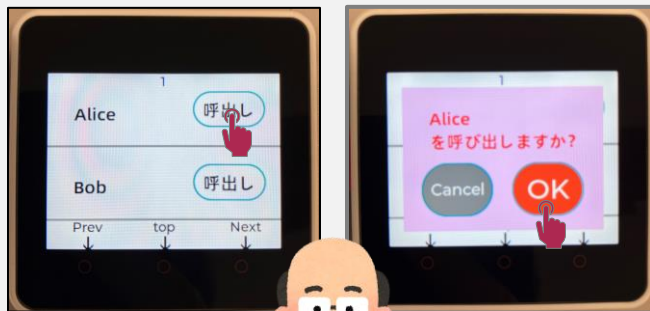
name	“yankee”	
SNS	 	 
	 	 
資格・試験	IPA (ES/SC/NW/DB/SA) PL-900、日商簿記2級	
最近の興味	ローコード全般、M5Stack、Azure、 Postman、Linux、仮想化などなど	



本日のアジェンダ

- 🗨️ つくってみたもの
- 🗨️ つかったもの
- 🗨️ 全体構成
- 🗨️ Azure Logic Apps
- 🗨️ M5Stack
- 🗨️ おわりに

つくってみたもの



※内容はフィクションで
十

つかったもの Azure Logic Apps

Azure Logic Apps は、コンテナ化されたランタイム上に構築された、主要なサービスとしての統合プラットフォーム (iPaaS) です。どこでも Logic Apps をデプロイして実行することができ、ビジネスクリティカルなワークフローをどこでも自動化しながら、スケールと移植性を向上させることができます。

公式HPの記載の一部を抜粋 ([ロジック アプリ サービス - iPaaS | Microsoft Azure](#))


- ☁ Azure上で使用できるiPaaS、操作感はPower Automateのクラウドフローと似てる
- ☁ 従量制なので、Power Automateでのプレミアムコネクタも使用できる (もちろん使った分だけお金はかかる)
- ☁ Power Automate Premium契約してなくて、ちょっとプレミアムコネクタ試したい場合は個人的に使い勝手が良い

参考記事

 @fsdg-takada in  株式会社船井総合研究所

Azure Logic Apps を触ってみる (簡単なデモ付き)

[Azure Logic Apps を触ってみる \(簡単なデモ付き\)](#)
[#初心者 - Qiita](#)

 @akihiro_suto (Akihiro Suto)

情シスよ、Azure Logic Appsをさわるんだ！

[情シスよ、Azure Logic Appsをさわるんだ！ #ポエム - Qiita](#)

つかったもの M5Stack

マイコンボードの一種

ブロックプログラミングやPython、
Arduino (C++ライク言語) でプロ
グラミングができる

M5Stackシリーズでいろんな種類
のマイコンやセンサユニットが販売

Maker界隈で人気で、色々な作品
が公開されている
(スタックチャンとかが有名)



[m5-docs \(m5stack.com\)](https://m5-docs(m5stack.com))



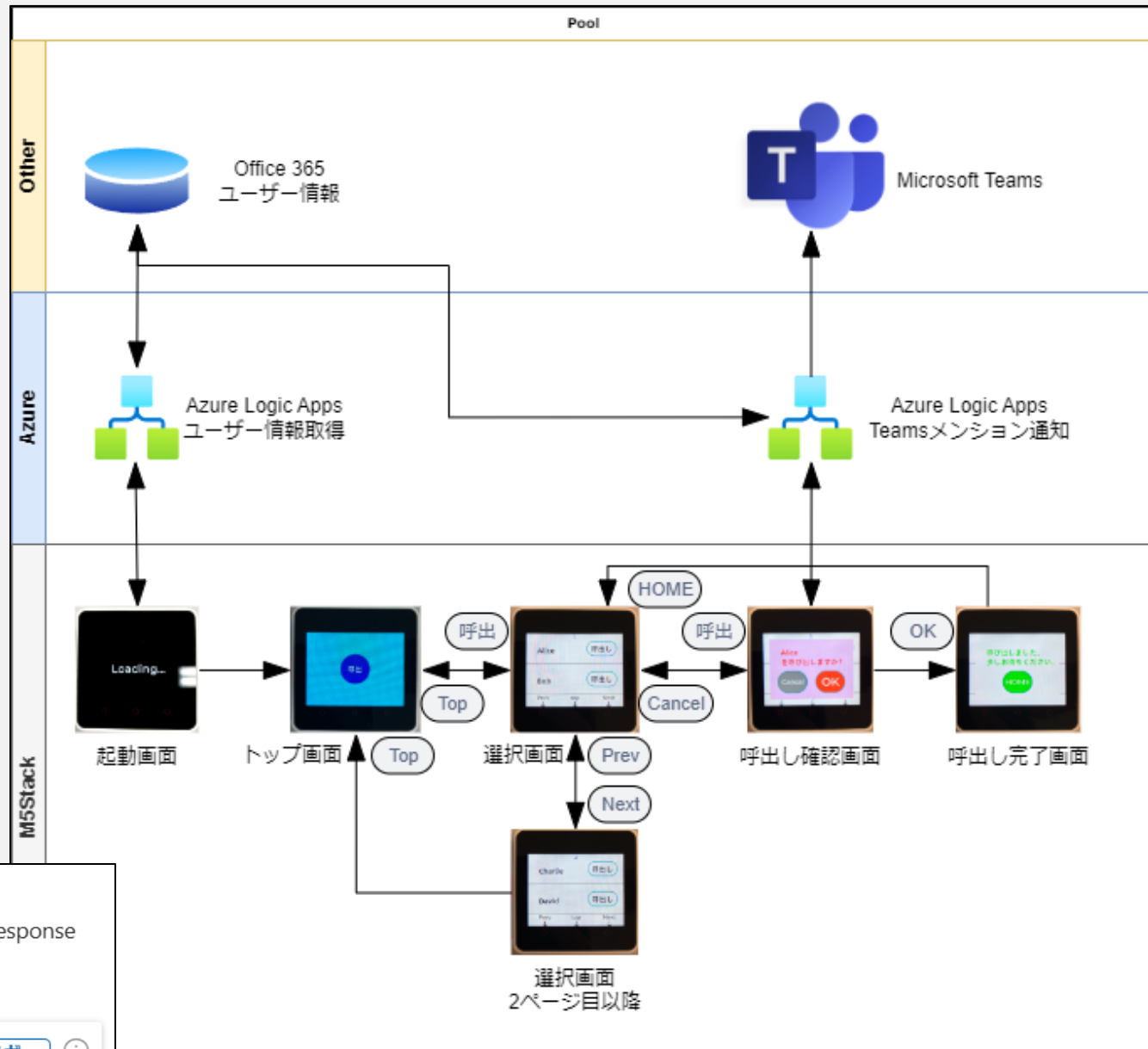
[スタックチャン \(Stack-chan\) | ProtoPedia](#)



[M5Stack Japan Creativity Contest 2024 | ProtoPedia](#)

全体構成

- 操作端末はM5Stack
- Azure Logic Apps経由で
 - ・ユーザー情報の取得
 - ・Teamsへのメンション通知を行うアプリをそれぞれ定義
- HTTPリクエストの受信をトリガとし、M5StackからHTTPリクエストする



Request

Operations to handle inbound request to workflow and send workflow response

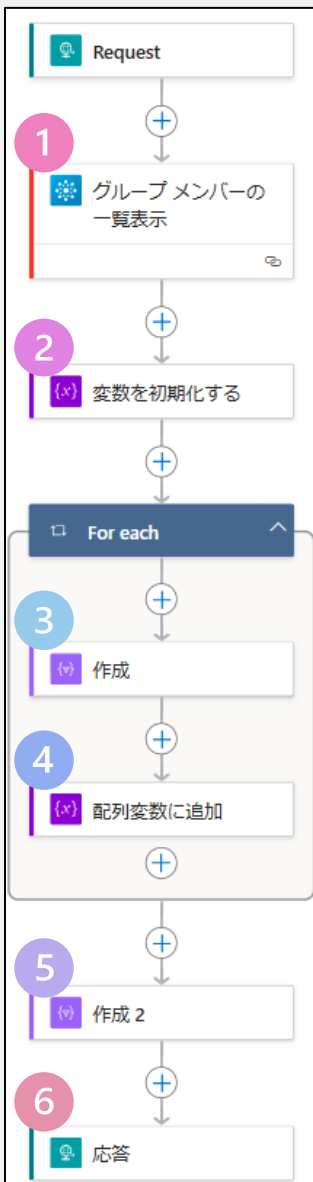
When a HTTP request is received

アプリ内

トリガー



Azure Logic Apps (ユーザ情報取得)



1

> グループメンバーの一覧表示

パラメーター 設定 コードビュー

グループ ID *

実験用

コネクタ
Office 365 Groups

操作メモ
この操作を実行すると、指定したグループ内のすべてのメンバーの一覧のほかに、名前、タイトル、電子メールなどの詳細情報が返されます。Top パラメーターを使用すると、1000 件までの項目をクエリできます。1000 件を超える値を取得する必要がある場合は、設定 -> 改ページの機能を有効にして、しきい値の制限を指定してください。
[詳細情報](#)

2

> 変数を初期化する

パラメーター 設定 コードビュー 情報

Name *

userList

Type *

Array

Value

Enter initial value

3

> 作成

パラメーター 設定 コードビュー 情報

Inputs *

```
{
  "UPN":  ユーザープリン... × ,
  "nickName":  表示名 ×
}
```

4

> 配列変数に追加

パラメーター 設定

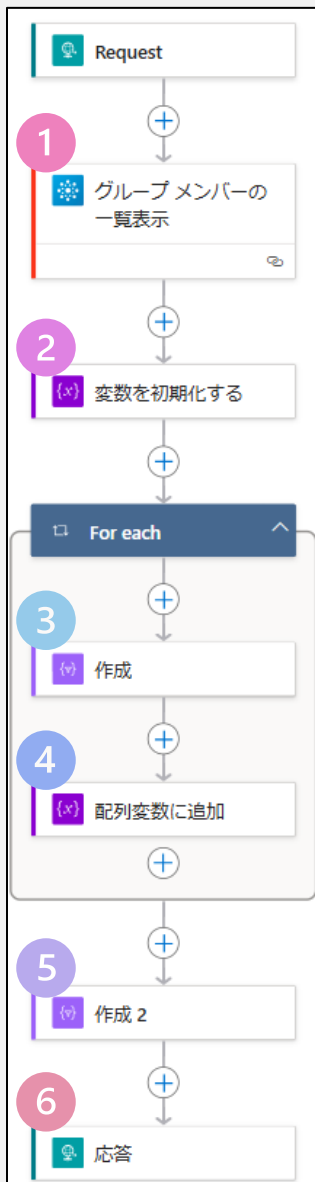
Name *

userList

Value *

出力 ×

Azure Logic Apps (ユーザ情報取得)



5

> 作成 2

パラメーター 設定 コードビュー

Inputs *

```
{  
  "employee": userList ×  
}
```

6

> 応答

パラメーター 設定 コードビュー 情報

Status Code *

200

Headers

キーの入力	値の入力	
		追加

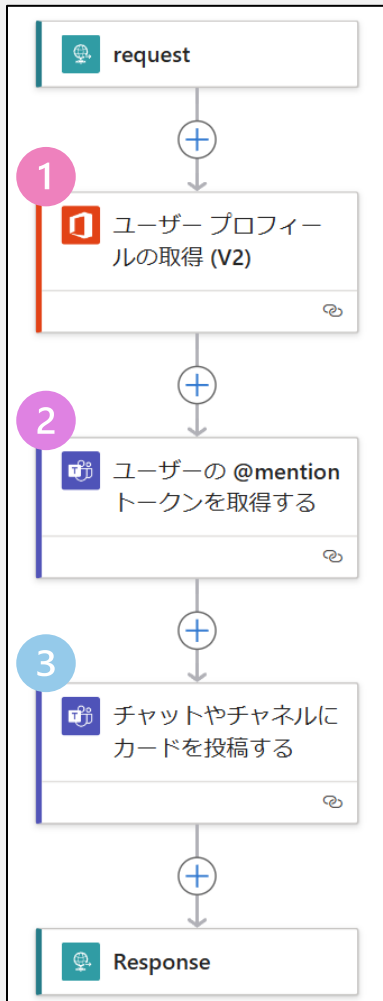
Body

出力 ×

出力例

```
{  
  "employee": [  
    {  
      "UPN": "Alice@***.com",  
      "nickName": "Alice"  
    },  
    {  
      "UPN": "Bob@***.com",  
      "nickName": "Bob"  
    }  
  ]  
}
```

Azure Logic Apps (Teamsメンション通知)



1

ユーザープロフィールの取得 (V2)

パラメーター 設定 Code view テスト 情報

ユーザー (UPN) *

UPN ×

詳細パラメーター

1 件中 0 件を表示しています すべてを表示

2

ユーザーの @mention トークンを取得する

パラメーター 設定 Code view テスト 情報

ユーザー *

ユーザープリンシパ... ×

3

チャットやチャンネルにカードを投稿する

パラメーター 設定 Code view テスト 情報

投稿者 *

フロー ボット

投稿先 *

Channel

Team *

実験用

Channel *

受付

Adaptive Card *

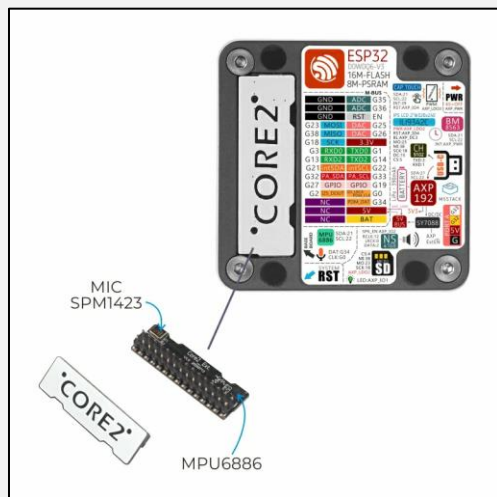
```
{
  "type": "AdaptiveCard",
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "version": "1.4",
  "body": [
    {
      "type": "TextBlock",
      "text": "🔔 部署 × @ニックネーム × へ",
      "wrap": true,
      "size": "Large",
      "color": "Attention",
      "separator": true
    }
  ],
  "items": [
    {
      "type": "TextBlock",
      "text": "@mention ×",
      "horizontalAlignment": "Right",
      "height": "stretch",
      "size": "Small",
      "color": "Dark"
    }
  ]
}
```

```
{
  "type": "TextBlock",
  "text": "🔔 部署 × @ニックネーム × へ",
  "wrap": true,
  "size": "Large",
  "color": "Attention",
  "separator": true
}
```

```
"items": [
  {
    "type": "TextBlock",
    "text": "@mention ×",
    "horizontalAlignment": "Right",
    "height": "stretch",
    "size": "Small",
    "color": "Dark"
  }
]
```

M5Stack

使用デバイス
M5Stack Core2

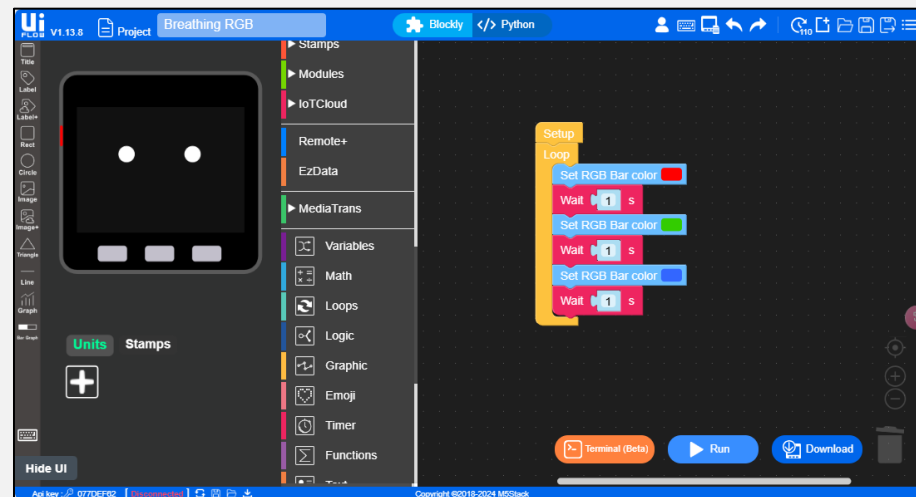


[m5-docs \(m5stack.com\)](https://m5-docs(m5stack.com))

📺 タッチスクリーン搭載

📶 Wi-Fi/Bluetooth内蔵

開発環境
UIFlow



[m5-docs \(m5stack.com\)](https://m5-docs(m5stack.com))

📺 ビジュアルプログラミング (Blockly)

📺 UIパーツの配置で画面描画ができる

M5Stack (ブロック全体)

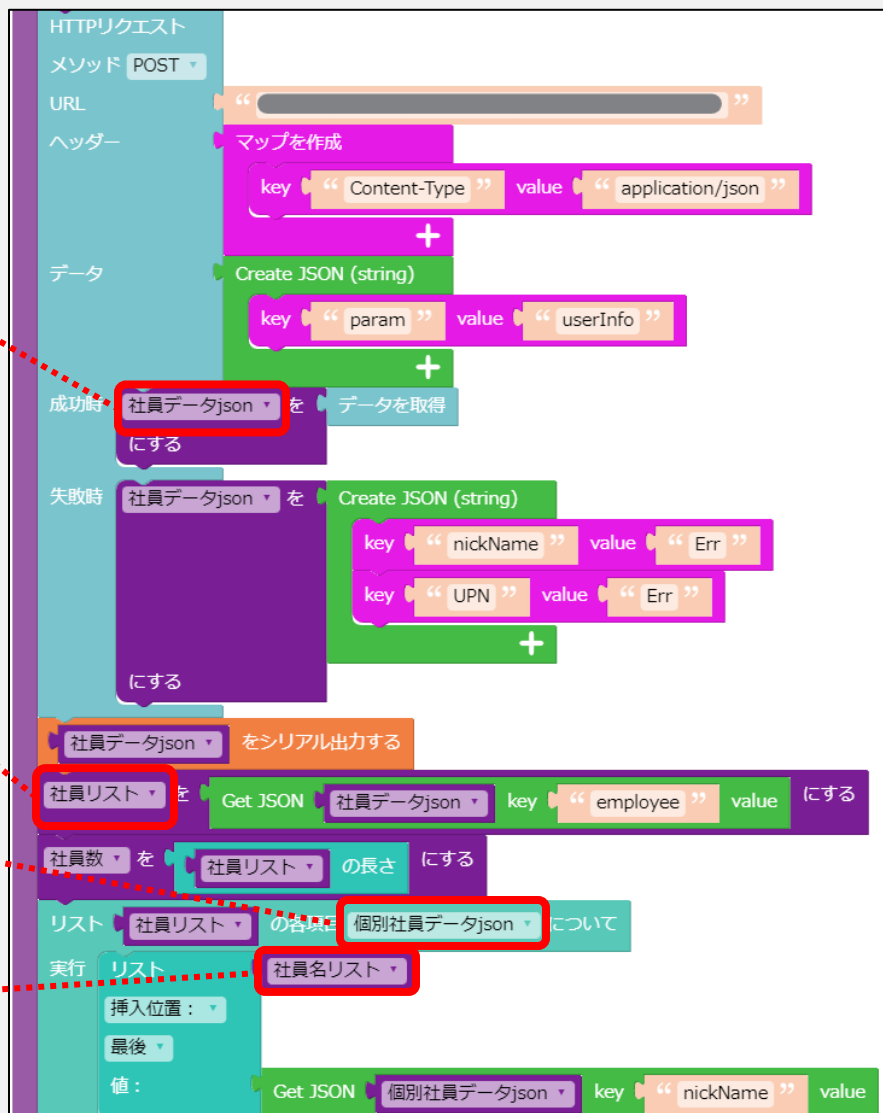


ブロック多いな・・・、これ全部説明したら5分じゃ終わらなくね？

~~5分でまとめきれなかったの、~~

今回はAzure Logic Appsとの連携部分のみ紹介します

M5Stack (ユーザ情報取得)



1

```
社員データjson
{
  "employee": [
    {
      "UPN": "Alice@***.com",
      "nickName": "Alice"
    },
    {
      "UPN": "Bob@***.com",
      "nickName": "Bob"
    }
  ]
}
```

2

```
社員リスト
[
  {
    "UPN": "Alice@***.com",
    "nickName": "Alice"
  },
  {
    "UPN": "Bob@***.com",
    "nickName": "Bob"
  }
]
```

3

```
個別社員データjson
{
  "UPN": "Alice@***.com",
  "nickName": "Alice"
}

{
  "UPN": "Bob@***.com",
  "nickName": "Bob"
}
```

4

```
社員名リスト
[
  "Alice",
  "Bob"
]
```

M5Stack (Teamsメンション通知)

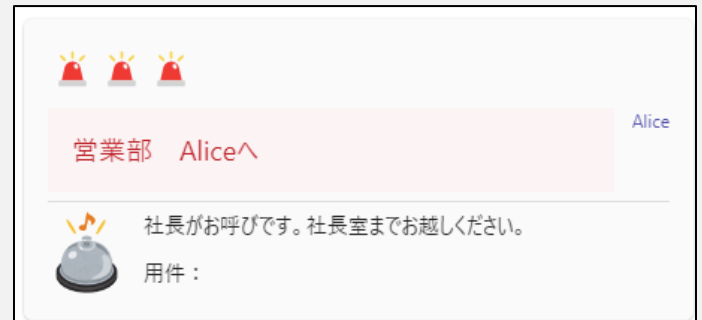
The image shows a Scratch script for sending a Teams notification. The script starts with an 'HTTP Request' block (method POST) and a 'Map creation' block for headers. The header 'Content-Type' is set to 'application/json'. The 'Data' block is set to '社員リスト' (Employee List) and is highlighted with a red box and a '2' in a pink circle. The script then branches into 'Success' and 'Failure' paths. In the success path, the first line of the result is set to '呼び出しました。' (Called) and the second line to '少しお待ちください。' (Please wait). In the failure path, the first line is set to 'エラー発生!' (Error occurred!) and the second line to an empty string.

1

```
社員リスト
[
  {
    "UPN": "Alice@***.com",
    "nickName": "Alice"
  },
  {
    "UPN": "Bob@***.com",
    "nickName": "Bob"
  }
]
```

2

```
データ
{
  "UPN": "Alice@***.com",
  "nickName": "Alice"
}
```




おわりに

- 💬 マイコンとiPaaSを使って呼び出しアプリを作ってみました
- 💬 iPaaSにマイコンを組み合わせることで、IoTっぽいことができます
- 💬 今回の内容が誰かの参考になれば幸いです

今回の内容は、Qiitaにも記事を載せていますので、詳細を知りたい方はご覧ください

 参考記事

 @yankee

M5Stack + Azure Logic Apps + Teamsを使った呼出しアプリ

[M5Stack + Azure Logic Apps + Teamsを使った呼出しアプリ #M5stack - Qiita](#)



Thank you!!

