



Azure Functionsを使ってSlack
に通知を試みよう

自己紹介

- 名前
 - 官野 慎太郎 (かんの しんたろう)
- 業務
 - 都内IT企業にてWeb系の開発業務に従事。
- 技術経験
 - Golang, JavaScript/TypeScript, MySQL, PostgreSQL, docker/kubernetes, etc...



目次

1. Azure Functionsとは
2. 今回使用する技術について
3. Slack Webhook URLの取得
4. Function Appの設定
5. ローカル上でのプログラム実装
6. Function Appへのデプロイ

Azure Functionsとは

- Azure(Microsoft社)が提供するサービスの一種。
- クラウド上にプログラムを実装することができる。
- 類似サービス：AWS Lambda, GCP Cloud Function

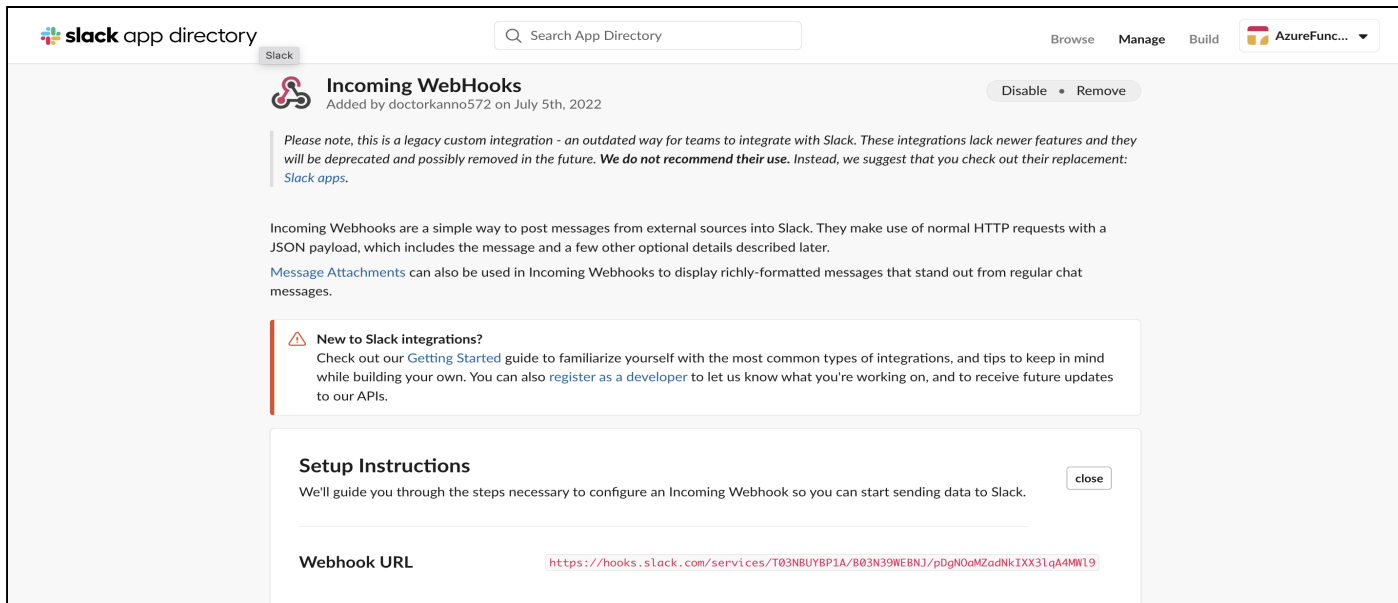


今回使用する技術

- Azure Functions Core Tools
- JavaScript
- Slack Webhook URL
- Weather Forecast API (Open-Meteo社)

Slack Webhook URLの取得

- Slack連携には`Incoming Webhook URL`の発行が必要
- `Slack app directory`の`Custom Integrations > Incoming WebHooks`より任意のWorkspaceのWebhook URL発行可能。



Incoming WebHooks より Web hook URLを作成

Function Appの設定

- Azure Portal上でFunction Appを構築
- 今回の名称は`azure-weather`
- 環境変数にSlack Webhook URLを設定

Create Function App ...

Basics Hosting Networking Monitoring Tags Review + create

Create a function app, which lets you group functions as a logical unit for easier management, deployment and sharing of resources. Functions lets you execute your code in a serverless environment without having to first create a VM or publish a web application.

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Resource Group * ⓘ [Create new](#)

Instance Details

Function App name *

azure-weather ✓

 .azurewebsites.net

Publish * ☒ Code ☐ Docker Container

Runtime stack *

Node.js

Version *

16 LTS

Region *

Japan East

i Not finding your App Service Plan? Try a different region or select your App Service Environment.

Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

Operating System * ☒ Linux ☐ Windows

Plan

The plan you choose dictates how your app scales, what features are enabled, and how it is priced. [Learn more](#)

Plan type * ⓘ

Linux Plan (Japan East) * ⓘ [Create new](#)

Sku and size *

Premium V2 P1v2
210 total ACU, 3.5 GB memory

ローカル上でのプログラム実装

- ローカルに空の`az-function`というディレクトリを作成
- `az-function`内で`func new`を実行
 - runtimeは`node`
 - 使用する言語は`JavaScript`
 - templateは`TimerTrigger`
 - Function Nameは`weather`

```
sintaro.kanno@kannonoMacBook-Pro az-function % func new
Select a number for worker runtime:
1. dotnet
2. dotnet (isolated process)
3. node
4. python
5. powershell
6. custom
Choose option: 3
node
Select a number for language:
1. javascript
2. typescript
Choose option: 1
javascript
Writing package.json
Writing .gitignore
Writing host.json
Writing local.settings.json
Writing /Users/sintaro.kanno/az-function/.vscode/extensions.json
Select a number for template:
1. Azure Blob Storage trigger
2. Azure Cosmos DB trigger
3. Durable Functions activity
4. Durable Functions entity
5. Durable Functions Entity HTTP starter
6. Durable Functions HTTP starter
7. Durable Functions orchestrator
8. Azure Event Grid trigger
9. Azure Event Hub trigger
10. HTTP trigger
11. IoT Hub (Event Hub)
12. Kafka output
13. Kafka trigger
14. Azure Queue Storage trigger
15. RabbitMQ trigger
16. SendGrid
17. Azure Service Bus Queue trigger
18. Azure Service Bus Topic trigger
19. SignalR negotiate HTTP trigger
20. Timer trigger
Choose option: 20
Timer trigger
Function name: [TimerTrigger] weather
Writing /Users/sintaro.kanno/az-function/weather/index.js
Writing /Users/sintaro.kanno/az-function/weather/readme.md
Writing /Users/sintaro.kanno/az-function/weather/function.json
The function "weather" was created successfully from the "Timer trigger" template.
```

対話形式で関数を作成

初期ディレクトリ構成

- 今回手を加えるのは`index.js`と`function.json`

```
| - .vscode
|   | - extention.json
| - weather
|   | - function.json ←★
|   | - index.js ←★
|   | - readme.md
| - .gitignore
| - host.json
| - local.setting.json
| - package-lock.json
| - package.json
```

実装部分

- `axios` と `@slack/webhook` を使用
- `func start` でローカル実行

```
const axios = require('axios');
const { IncomingWebhook } = require('@slack/webhook');

module.exports = async function (context, myTimer) {
  // ~~~~~ 以下実装部分 ~~~~~
  const response = await axios.get('https://api.open-meteo.com/v1/forecast?latitude=35.6785&longitude=139.6823&hourly=temperature');

  const webhook = new IncomingWebhook(process.env.WEBHOOK);

  await webhook.send({
    text: `現在の気温は${response.data.current_weather.temperature}です。`,
    channel: "#demo",
    username: "Azure Functions",
  });
  // ~~~~~ 実装部分ここまで ~~~~~
};
```

Function Appへのデプロイ

- `function.json`を編集
 - 実行間隔を5分おきに設定
 - `scriptFile`を`index.js`に指定
- `package.json`を`weather`に移動
- `azure-weather`に`weather`ディレクトリをデプロイ
 - 実行コマンド: `func azure functionapp publish azure-weather`

function.json内部

```
{
  "bindings": [
    {
      "name": "myTimer",
      "type": "timerTrigger",
      "direction": "in",
      "schedule": "0 */5 * * * *"
    }
  ],
}
```

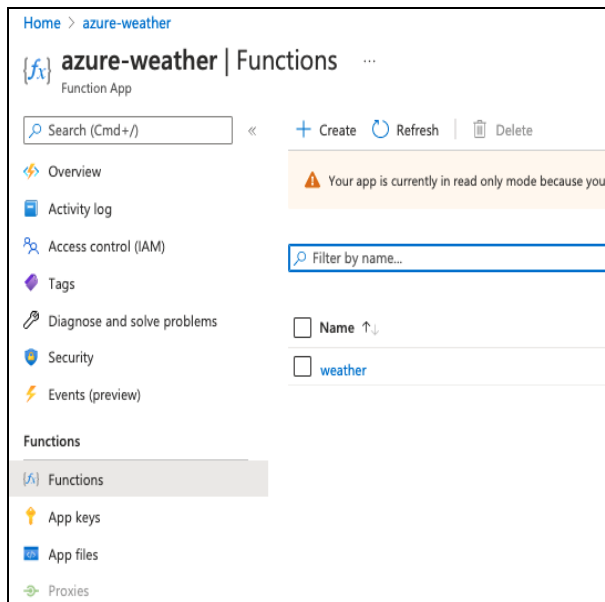
weather内の最終的な構成

```
| - weather
    | - function.json
    | - index.js
    | - readme.md
    | - package.json ← 上の階層から移動
```

デプロイ後のazure-weather

- Functionsに`weather`が存在
- 5分おきに現在の気温をSlackへ通知

デプロイされたweather



Slackへの気温の通知

