

Azure Functionsとは、Microsoft Azure（以下、Azureといいます。）で提供されている、アプリケーション開発関連サービスの1つになります。サーバーレスアーキテクチャが採用されており、マイクロサービスを実装していくことができます。

「サーバーレスアーキテクチャってなんだろう？」「マイクロサービスってどういうこと？」と思われる方も多いと思います。これらは、クラウドサービスが発展してく上で今後主流になって行く重要な概念になります。

本コラムでは、これらの概要が理解できるように解説させていただきます。

## 目次：

Azure Functionsとは？

サーバーレスアーキテクチャ

マイクロサービス

トリガーとバインド

開発言語

Azure Functionsのプラン

従量課金プラン

Premiumプラン

App Serviceプラン

ローカル環境でコーディングしてテストする

ローカル開発環境

まとめ

# Azure Functionsとは？

Azure Functionsは、Azure上でコード（関数）をサーバーなしに（サーバーレスで）実行できるサービスです。

利用者は、サーバーを意識する必要がなく、アプリケーションロジックの開発に集中することができます。

Azure Functionsを利用することによって、WebAPIやWebサービスを効率的に作ることができます。一つのフレームワークと言い換えた方が分かり易いかもしれません。

## サーバーレスアーキテクチャ

「サーバーレス」といっても実際にサーバーが存在しない、サーバーが要らない。というわけではありません。サーバーの存在を意識することなく利用することができる、ということです。サーバーレスアーキテクチャはクラウドサービスが目指している近い将来のあるべき姿の一部がいち早く具現化されているサービスと言えます。

Azure Functionsを利用すると、フルマネージドサービスとしてサーバーなどの開発環境が提供されますので、アプリケーションに大量のリクエストが来た場合への対応策の一つとしてサーバー台数を増やすことなどを気にする必要はありません。Azureが自動的に対応してくれます。また、サーバーのハード障害についても心配する必要がありません。自動的に切り替えてくれます。

ユーザーは、サーバー構築や障害への対応を気にすることなく、ただアプリケーションのロジックを書くことに集中できるようになります。

## | マイクロサービス

マイクロサービスとは、開発システムを、メンテナンス性を高めることを目的として、小さなモジュール群で構成されるサービス開発手法です。なお、マイクロサービスは開発手法の呼び名であり、どのような技術を実装で利用するかについては問われません。

## | トリガーとバインド

Azure Functionsを利用するにあたり、気にすべきことは作成したいアプリケーションロジックに対応しているかです。Azure Functionsでは、「トリガー」を使ってアプリケーションを起動することが出来ます。

また、アプリケーションによって導き出された結果を「バインド」を利用することで引数の受け渡しが可能になっています。

### トリガーとして利用できるバインディングとしては次のものがあります

HTTP Trigger : HTTP要求をトリガーにする

Timer : 時刻を指定してコードを起動します。CRON相当のスケジュール管理ができる

GitHub webhook : GitHubリポジトリで発生するイベントをトリガーにする

汎用Webhook : 外部Webサービスのwebhookをトリガーにする

Cosmos DB : Azure Cosmos DBへのデータ追加及び更新されたことをトリガーにする

Blob Storage : Azure Storage BLOBがコンテナに追加されたことをトリガーにする

Queue Storage : Azure Storageキューにメッセージが配信されたことをトリガーにする

Event Hubs : Azure EventHubに送信されたことをトリガーにする。IoTで使いやすいシナリオ

Service Bus Queue : メッセージ・キューをリッスンして、コードを起動する

Service Bus Topic : トピックをサブスクライブすることで、コードを起動する

## Microsoft Graphイベント

# Outputに利用できるバインディングには上記に加え、以下があります。

Microsoft Excelテーブル

Microsoft Graph OneDrive ファイル

Microsoft Graph Outlookメール

Mobile Apps

Send Grid

Service Bus

SignalR

Table Storage

Twilio

## 開発言語

Azure Functionsでは開発言語は、C#, Java, JavaScript, Python, PowerShellのいずれかの言語を自由に選んで関数を記述できます

# Azure Functionsのプラン

Azure Functionsには、次の3種類の価格プランがあり、利用するニーズに最適なプランを選択します。選択するプランにより、次のような動作に影響があります。

関数アプリをスケールする方法

各関数アプリをインスタンスに利用できるリソース

Azure Virtual Network接続などの高度な機能のサポート

## 従量課金プラン

リソース（サーバー等）の管理について考慮する必要がなく、コードを実行した時間に応じた料金のみを支払います。仮想マシンを利用する時のようにアイドル状態の仮想マシンの料金や、予約容量の事前の料金を払う必要がありません。ただし、次のような制限があります。

タイムアウト時間が短いため、長時間の処理は実施できない

コールドスタートとなった場合に起動時に時間が掛かる（処理を開始するのが遅くなる）

# Premiumプラン

常にオンラインの状態ですぐに回答する準備が整った、事前にウォーミングされたインスタンスをいくつか自分で指定します。関数が実行されると、追加で必要となるコンピューティングリソースがAzureから提供されます。

Azureによってアプリケーションがスケールアウトまたはスケールインされたときに追加で使われるインスタンスが、支払いの対象になります。

# App Serviceプラン

関数をWebアプリと同様に実行できます。Azure App Serviceを他のアプリケーションに使用している場合、追加コストを負担することなく、同じプランで関数を実行できます。

従量課金プランでのデメリットを解消するために、このプランを選択します。料金は少し上がってしまいますが。

稼働させたいアプリケーションの特徴によって、最適なプランを選択することにより、管理負荷とコストを最適化しましょう。

## ローカル環境でコーディングしてテストする

Azure Functionsの開発やテストは、Azure Portalから行うことができますが、多くの開発者は、ローカルコンピュータ上での開発を好むようです。Azure Functionsでは、ローカルコンピュータ上でお気に入りのコードエディターや開発ルーツを使用して、関数を作成し、テストすることができます。

ローカル関数はライブAzureサービスに接続できるため、完全なFunctionsランタイムを使用してローカルコンピュータ上で関数をデバックすることができます。

# ローカル開発環境

ローカルでの環境は、使用する言語とツールによっても異なりますが、基本的には次のツールを利用することが可能です。

Visual Studio Code

コマンドプロンプトまたはターミナル

## Visual Studio 2019

### Maven

ローカルコンピュータ上で開発するときの最大の注意事項は、ローカル開発とポータル開発を混在させてはいけない、ということです。ローカルプロジェクトから関数を発行するときは、ポータルではプロジェクトコードを管理して変更しないようにしないといけません。

# まとめ

Azure Functionsは、ユーザーが開発環境としてのサーバー構築や運用のことを気にすることなくアプリケーション開発できるようにするための仕組みです。コード以外の部分がすべてフルマネージドサービスとして提供されるAzure Functions。サーバーレスアーキテクチャやマイクロサービスアーキテクチャはこれからの技術として注目されておりますが、その素晴らしさの片鱗を垣間見ることができるのではないのでしょうか。まずは業務アプリケーションの一つの小さな機能からでも、サーバーレス化を進めてみて、感触をためさせてはいかがでしょうか。