

2019年1月22日
株式会社インプレスR&D
<https://nextpublishing.jp/>

JavaScript 初心者でもできる！簡単サーバーレスサービス
『Netlify で始めるサーバーレス開発』発行
技術書典シリーズ、1月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレスR&Dは、『Netlify で始めるサーバーレス開発』（著者:竹本 雄貴）を発行いたします。

『Netlifyで始めるサーバーレス開発』
<https://nextpublishing.jp/isbn/9784844398738>



著者:竹本 雄貴
小売希望価格:電子書籍版 1600円(税別)／印刷書籍版 1800円(税別)
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様:B5判／カラー／本文80ページ
ISBN:978-4-8443-9873-8
発行:インプレスR&D

<< 発行主旨・内容紹介 >>

【JavaScript 初心者でもできる！Netlify で作るサーバーレスサービス】

本書は JavaScript やサーバーレスの初心者に向けた入門書です。処理内容だけを書くことでサーバーレス開発ができる FaaS のひとつ「Netlify」を利用しつつ、Functions の利用方法、LINE Bot や Slack App の開発方法を取り上げつつ、FaaS を活用した開発を学習し、そのメリットが実感できる内容となっています。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

FaaS と一般的なアプリケーションの違いを丁寧に紹介

図2.1: FaaS と一般的なアプリケーションの比較

FaaSの場合
リクエスト → FaaS (実現したい処理) → レスポンス
管理する必要があるコード

自前実装の場合
リクエスト → アプリケーション (実現したい処理) → レスポンス
管理する必要があるコード

Node.jsで同じようなものを書く必要があるとすれば、expressでリクエストを受けて、URLに応じて適切な関数を実行させる必要があります。すでにNode.jsサーバーがあったら、書き足すだけでいいじゃないか?と思う方もいるかもしれませんが。しかし、書き足し続けると依存パッケージが増えてデプロイ時間が増えたり、依存パッケージの競合で昔の関数が壊れたりするなどの弊害が出てきます。

FaaSであれば関数単位で作るので、依存関係が少ないまま開発できます。

Netlifyを選ぶ理由

筆者がなぜ数あるFaaSの中からNetlifyでのFaaSの入門を推しているかですが、Netlifyは「始めやすいから」の一点に尽きます。

FaaSとして有名なものはAWSのLambdaとGCPのCloud Functionsですが、これらの利用にはAWSやGCPの登録が必要で、無条件で利用されていますが、他の用途で使っていた場合に意図しない課金が発生する場合があります。そもそも管理画面が複雑で利用までのハードルが高いと捉えます。

一方、Netlifyはほぼ無料で使うことができ、課金するときの管理画面も明確です。

また、NetlifyのGitHub連携は素晴らしい、ブランチやプルリクエストごとにプレビュー環境が立つので快適に開発を進めることが可能です。AWSやGCPであれば、CircleCI/TravisCIなどで別途デプロイ処理を書く必要があるでしょう。これらの理由から、筆者はNetlify Functionsの利用を薦めています。

2.2 Functionsにハローワールド

それではNetlify Functionsを実際に使ってみましょう。

Functionsを使うには、GitHubでの静的サイトのホスティングと同様に、GitHubのリポジトリと連携して利用します。

静的サイトのホスティングとの差としては次の2点があります。

```
netlify.toml
[build]
  functions = "functions"
```

次に利用したい関数を作成します。JSファイルを functions/hello.js として作成しましょう。

```
functions/hello.js
exports.handler = function(event, context, callback) {
  callback(null, {
    statusCode: 200,
    body: "Hello, World"
  });
}
```

Netlify Functionsでは、exports.handler で export した関数を Netlify の Functions として実行できます。

関数の引数 event, context, callback は event にリクエストの情報が入っており、callback で関数の出力を調整するものという認識で問題ありません¹⁾。なおこの形式はAWSのLambdaと同一なので、Google等で調べるときには「Lambda」と検索するのもよいでしょう。

この時点で functions/hello.js と netlify.toml のふたつのファイルができました。GitHubにpushして、対応するリポジトリをNetlifyと紐づけましょう。

この状態でNetlifyの対応するページを見て、Functionsタブを選択してみましょう。図22のようにhellojsという記述があればFunctionsのデプロイは成功です。

1) 個人で関数の書き方を確認しています

20 | 第2章 Netlify Functions入門

第2章 Netlify Functions入門 | 21

LINE や Slack の Bot 開発を通じて Netlify の機能を学習

第3章 LINE Bot開発

第2章ではNetlify Functionsの利用方法を解説しました。本章では実践的な題材としてLINEのBotを取り上げます。最初に「きまった内容を送るBot」を作り、応用として「LINE Botで完結するシミュレーション」を作っていきます。

3.1 LINE Botの仕組み

LINE BotのメッセージはMessaging APIというAPIを通して送信できます。その中でReply APIとPush APIというメッセージの送信手段が提供されています。

Reply APIは読んで字のごとく、ユーザーのメッセージに対して返信を行うAPIです。ユーザーがメッセージを送信した際に、開発者が指定したURLにリクエストが送信されます。受け取ったリクエストにReplyTokenというトークンが付与されているので、そのトークンと返信したいメッセージを使って、指定のエンドポイントにPOSTリクエストを行うことでメッセージが送信できます。

一方のPush APIは、ユーザーのメッセージと関係なくメッセージを送信できるAPIです。Reply APIと違い利用のハードルが高めで、Push APIを利用するには2018年12月現在、LINEの「プロプラン（月額21,600円）」に加入する必要があります。そのため個人で利用するのは難しいかもしれません。

送信するためには、送信したいメッセージを付与してPush用のエンドポイントを叩くだけです。利用する際はLINE Developersのドキュメントを参照してください。

本書では無料で使えるReply APIを使って開発を進めていきます。

3.2 Developer登録とチャネル登録

LINEのAPIを利用するには、まず開発者登録を行う必要があります。また、利用の際にチャネルというアプリケーション情報のようなものを登録する必要があります。

LINE Developersより自分のLINE IDでログインしてください。

LINE Developers
https://developers.line.me/ja

ログインするとプロバイダ選択画面が表示されます（図3.1）。プロバイダは自分のアカウントを選択しましょう¹⁾。

1) Facebookに紐付いた個人アカウント・グループアカウントの選択はできません。

30 | 第3章 LINE Bot開発

第3章 LINE Bot開発 | 31

付録A Functionsの便利イディオム

本章では、関数を実装するときによく検索したくなる情報をまとめました。ぜひ手元に置きながら開発を進めてみてください。

A.1 event引数の中身

関数を実装するときによく検索するevent引数の情報です。次のコードをsrc/args.jsに実装した場合の出力を紹介します。contextはランタイム情報¹で、callbackはレスポンス用の関数なので、一番使用頻度の高いeventの中身を取り上げます。

```
src/args.js
exports.handler = function(event, context, callback) {
  console.log(JSON.stringify(event, null, 4))
  callback(null, {
    statusCode: 200,
    body: "pong",
  });
}
```

GETリクエスト

https://[APP_NAME].netlify.com/.netlify/functions/args?hoge=aa&huga=11 へのリクエスト時のeventの中身です。

GETリクエストでは、URLのクエリパラメータがqueryStringParametersとなっているのがポイントです。

```
{
  "path": "/.netlify/functions/args",
  "httpMethod": "GET",
  "headers": {
    "accept": "",
    "accept-encoding": "",
    "accept-language": "",
    "client-ip": ""
  }
}
```

¹ https://docs.aws.amazon.com/ja_jp/lambda/latest/dg/rooksie-org-model-context.html
² headers (リクエストヘッダー) の中身は省略しています。

```
"connection": "",
"upgrade-insecure-requests": "",
"user-agent": "",
"via": "",
"x-bb-ab": "",
"x-bb-client-request-uuid": "",
"x-bb-ip": "",
"x-bb-loop": "",
"x-bb-country": "",
"x-forwarded-for": "",
"x-forwarded-proto": "",
"x-language": ""
},
"queryStringParameters": {
  "hoge": "aa",
  "huga": "11"
},
"body": "",
"isBase64Encoded": true
}
```

POSTリクエスト

POSTで送るケースは、JSONでデータを送る場合とFormData³としてデータを送る場合があります。

GETリクエストの例と同じく、https://[APP_NAME].netlify.com/.netlify/functions/args へのリクエスト時のeventの中身を紹介します。

1. JSONでデータを受け取るケース

リクエストボディの中身

```
{
  "fuga": "bb",
  "hoge": "aa"
}
{
  "path": "/.netlify/functions/args",
  "httpMethod": "POST",
  "headers": {
    "accept": ""
  }
}
```

³ Content-Type(application/www-form-urlencoded, multipart/form-data)の値が設定されています。

<< 目次 >>

第1章 Netlify 入門

- 1.1 はじめてのNetlify
- 1.2 Netlifyとは?
- 1.3 サーバーレスアプリケーションとは

第2章 Netlify Functions 入門

- 2.1 FaaSとは
- 2.2 Functions にハローワールド
- 2.3 netlify-lambda を使った関数
- 2.4 Netlify Functions の用途
- 2.5 Netlify Functions とAWS Lambda

第3章 LINE Bot 開発

- 3.1 LINE Bot の仕組み
- 3.2 Developer 登録とチャンネル登録
- 3.3 Webhookを受け取る
- 3.4 Reply API でメッセージに返信する
- 3.5 LINE Bot 開発実践

第4章 Slack App 開発

- 4.1 事前準備
- 4.2 メッセージのフォーマット
- 4.3 Slash Commands

- 4.4 Events API の利用
- 4.5 Web API の利用
- 第5章 この本の後に取り組むべきこと
- 5.1 Netlify でのサーバーレス開発
- 5.2 Netlify 以外のサーバーレス開発
- 付録A Functions の便利イディオム
- A.1 event 引数の中身
- A.2 逆引きシチュエーション
- 付録B TypeScript 対応
- B.1 ボイラープレートの紹介

<< 著者紹介 >>

竹本 雄貴(たけもと ゆうき)

「つのぶえデザイン」の屋号で Web 開発を中心にを行うフリーランスエンジニア。スタートアップや中小企業を中心に開発支援・技術相談・UI デザインを行っている。多くの新規サービス開発案件に携わって行く中で、立ち上げ速度の早い開発手法を重視するようになった。Netlify と相性がよい静的サイトジェネレーターである Gatsby のメンテナー。Nuxt.js や ReactNative、Netlify 系の OSS にもコントリビュートしている。

Blog: <https://mottox2.com>

GitHub: @mottox2

Twitter: @mottox2

note: <https://note.mu/mottox2>

<< 販売ストア >>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D (本社: 東京都千代田区、代表取締役社長: 井芹昌信) は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社: 東京都千代田区、代表取締役: 唐島夏生、証券コード: 東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp