

2018年4月10日
株式会社インプレスR&D
<https://nextpublishing.jp/>

JavaScript 初心者でもできる！Node.js と SQL でつくる Web アプリ
『Azure 無料プランで作る！初めての Web アプリケーション開発』発行
「技術書典シリーズ」最新刊

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、『Azure 無料プランで作る！初めての Web アプリケーション開発』（著者：窓川 ほしき）を発行いたします。

『Azure無料プランで作る！初めてのWebアプリケーション開発』
<https://nextpublishing.jp/isbn/9784844398219>



著者：窓川 ほしき
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 114 ページ
ISBN：978-4- 8443-9821-9
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！Web アプリ簡単作成ガイド！】

本書は JavaScript 初心者がすべて Azure の無料枠を使って実際に Web アプリケーションを作ること为目标としたチュートリアルガイドです。Node.js と SQL を組み合わせて、ブラウザベースの Web サービスを構築できるまでを丁寧に解説しています。JavaScript 初学者に最適な一冊です。

(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。)

第2章 Azureの環境を準備して、スクレイピングアプリを公開する

本章では、RESTful APIのWebサービス公開場所として使うためのAzureのアカウント作成と公開設定を行います。その際にソースファイルのデプロイ元としてGitHubサービスのアカウントを紐付けます。

- 次の流れでアカウントの作成と設定を行います。
1. GitHubアカウントの作成
 2. GitHubにリモートリポジトリを作成
 3. GitHubのローカルリポジトリを作成してファイルを格納
 4. Azureアカウントの作成
 5. Azure Web APPのインスタンス作成
 6. Azure Web APPにGitHubリポジトリを紐付け
 7. Azure Web APPの環境変数などを設定して動作確認

2.1 GitHubアカウントとリポジトリの作成

GitHubアカウントを作成します。GitHubのトップページにアクセスして、メールアドレスとパスワードを設定するだけで作成することができます。有料プランと無料プランがありますが、本書の内容を試す範囲では無料プランを選択して問題ありません。

GitHubとは

- 次のサイト様の解説が分かりやすいので参照ください。
- ・今さら聞けない！GitHubの使い方【超初心者向け】
<https://techacademy.jp/magazine/6235>
 - ・Pro Git 日本語版電子書籍公開サイト
<https://progit-jp.github.io/>

GitHubで使われる用語については、簡単には次のように捉えてください。

- ・リポジトリ
ファイルやディレクトリの状態を保存する場所（本書では「Azureへのソースファイル

- ・「中継場所」としてのみ利用します)。
- ・リモートリポジトリ
GitHubサービスの中核。ファイルと変更履歴の保管場所。
- ・ローカルリポジトリ
リモートリポジトリから、クローンされたローカルのリポジトリ。作業場所。
- ・クローン (Clone)
リモートリポジトリの状態をそのままそっくりローカルに再現する操作。
- ・コミット (Commit)
ローカルリポジトリに対して変更を記録する操作。
- ・プッシュ (Push)
ローカルリポジトリの変更結果を全てリモートリポジトリに反映する操作。
- ・プル (Pull)
リモートリポジトリの更新内容を、ローカルリポジトリに反映する操作。
- ・Sync
GitHub Desktop (GitHubサービスの公式クライアントアプリ) 上でのPush and Pull操作の事。

GitHubのより詳細な使い方については、書籍「わかばちゃん和学がGit使い入門」が分かりやすいのでお勧めいたします。

GitHubアカウントを作成したら、リポジトリを作成します。GitHubのトップページにアクセスしてログインします。「Your repositories」のカラムにある「New repository」のボタンを押します。新規リポジトリの作成画面図21が出るので、任意のリポジトリ名称を入れます。「Initialize this repository with a README」にチェックを付けます。「Add .gitignore: Node」を選択します。これにより、「node_modules」等の「Gitリポジトリに反映する必要が無いファイル群」を管理対象外にする設定を自動生成することができるので楽です。「Add license」の欄は任意に選んでください(筆者はMIT licenseにすることが多いです)。

バッテリー残量を記録するアプリを実際に作りながら応用を説明

第3章 バッテリーを記録して、マルチデバイスから参照できるアプリを作る

前章のアプリは、データはその場で生成して参照する形式の物でした。本章では、データをAzure側に保存する機能を作ります。具体的には、クライアントPCのバッテリー残量をクラウド(Azure)に保存しておき、それを複数の端末から参照できるようにします。本章で扱うサンプルコードは、次のような2つの機能からなります。

- ・クライアントPCからバッテリー残量の記録を受け付けて、サーバー側で記録する。
- ・任意のクライアント端末(PC、スマホ、その他)のブラウザから参照を受け付けて、サーバー側に保存してあるバッテリー残量ログを返却する。

記録対象のPCと、記録を参照する複数の端末クライアントのブラウザとの双方からデータの保存先であるデータベースが置かれたサーバーへアクセスする方法は、前章までと同様にRESTful APIを採用します。

3.1 データの保存先に、SQLデータベースを選択

データの保存先として、「SQLデータベース」を採用します。独自形式のファイルに保存しても良いのですが、参照はデータ数の管理の面から、一般に使われている「SQLデータベース」を利用した方が楽です。

データベースにも種類が様々ありますが、「利用者が多くて汎用性がある」というメリットから、「SQL言語でWrite/Readできる、関係データベース管理システム(RDBMS)」を選択します。「関係データベース」は「Excelの一行の単位で、情報を紐づけて記録する」とイメージしてください。「管理システム」は「ファイルシステム」くらいの理解が良いです。

一般に、関係データベース管理システムへのWrite/Readには「SQL言語」を用いることができます。したがって「SQL言語で、データベースへ保存と読み出しを行う」として、アプリを設計しておく、保存先のデータベースの種類(MS SQL Server, My SQL, SQLiteなど)に依存せず、その運用コストや動作スベックに応じて任意に選ぶことができます。また後で変更することも容易です。

本章でアクセスするデータベースを、以降では「SQLデータベース」と呼ぶことにします。

3.2 SQL Server Expressのインストールと動作確認

Microsoft SQL Server Express (以降、ローカルのSQL Serverと略記)と、Microsoft SQL Server Management Studio Express (以降、SS Management Studioと略記)をローカルのWindows環境にインストールします。本章のサンプルコードは、「SQL Server 2017 Express エディション」と「SQL Server Management Studio 17.5」で動作確認を行っています。ローカルのSQL Serverインストール時の設定項目は特にありません。デフォルトのままで「インストール」を押して進めてください。インストールはデフォルトの「SQL EXPRESS」が設定され、図31のように完了します(パソコンを再起動してください)。続いて、SS Management Studioをインストールします。こちらも特に設定項目はありません。最初の画面図32で「インストール」ボタンを押すだけです。

図3.1 ローカルのSQL Serverインストール完了画面



1. この方のURLからSQL Server 2017 Express Editionをダウンロードする。 <https://www.microsoft.com/ja-jp/sql-server/sql-server-edition-express>
 2. この方のURLからSSMS Setup-ENU.exeをダウンロードする。 <https://docs.microsoft.com/ja-jp/sql-server/download-sql-server-management-studio-express>
 3. SQL Server ExpressとSSMSのインストール方法については、こちらのWebページが参考になります。SQL Server 2016版ですが、SSMS 2017版でも同様の手順が適用できます。 <https://www.teknohive.com/2016/08/express/>

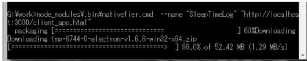
付録C Windows向けにElectronでネイティブアプリ化する

Windows 向けのアプリケーション提供を Electron で Windows 向けのネイティブアプリ化する方法を説明します。作業用に任意のフォルダを作成してください。作成したフォルダに移動し、コマンドラインから「`npm install nativefier`」を実行します。これにより `nativefier` モジュールがインストールされます。2次のコマンドを実行します。紙面の都合上2行で書いていますが、1行で記述してください。

```
node_modules\.bin\nativefier.cmd
--name "アプリケーション名" "Azureに公開したWebアプリの表示用のURL"
```

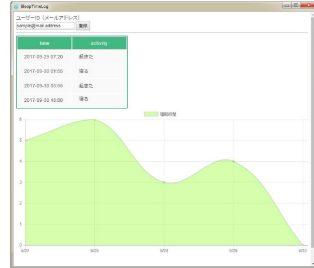
遷移画面が表示され(図C.1)、しばらくすると「アプリケーション名-win32-x64」というフォルダが生成されます。フォルダの中に「アプリケーション名.exe」が格納されています。この実行ファイルを実行すると、WebアプリをChromeブラウザで表示した時と同じ画面が表示されます(図C.2)。WebアプリのURLを聞くだけの専用ブラウザが出来ました。本フォルダ配下一式を配布することで、スマホ端末上からの操作と同様に、ブラウザを意識せずにパソコン上からも利用することができます。

図C.1: Electronベースのアプリ生成中スクリーンショット



1. 以下の作業ではGitHubリポジトリへのコミットは行いません。適宜nodeモジュールをインストールするたため部分フォルダでの作業を推奨しますが、同じフォルダで作業を進めることを推奨します。
 2. nativefier は従来のWebページをデスクトップアプリ化するためのツールです。真面目には、従来のWebページを動くだけの専用アプリも、Electronベースで生成してくれるツールです。
 3. nativefier をローカルにインストールしているため、node_modules 配下の実行ファイルも参照されている状態にしてください。グローバルにインストールする場合は、単に「`nativefier`」コマンドだけで実行が可能です。
 4. ここでは、Windows 向けに生成しますが、Mac向けやLinux向けに生成することも、同様に利用することの可能です。

図C.2: Electronベースでネイティブアプリ化した表示例



<<目次>>

第1章 スクレイピングアプリをローカルで作る

- 1.1 クライアントの表示UIとしてカレンダーを作成する
- 1.2 カレンダーに、追加の情報を表示する
- 1.3 ローカルで、動的な表示を行ってみる
- 1.4 ローカルでのNode.js のインストールとサンプルコードの動作確認
- 1.5 Twitter から特定のキーワードをスクレイピングする
- 1.6 スクレイピングしたデータをカレンダーに表示する

第2章 Azure の環境を準備して、スクレイピングアプリを公開する

- 2.1 GitHub アカウントとリポジトリの作成
- 2.2 Microsoft Azure アカウントの作成
- 2.3 Web サービスのリソース作成とGitHubリポジトリの紐付け
- 2.4 認証情報を設定する
- 2.5 さあ、複数のデバイスからアクセスしよう

第3章 バッテリーを記録して、マルチデバイスから参照できるアプリを作る

- 3.1 データの保存先に、SQL データベースを選択
- 3.2 SQL Server Express のインストールと動作確認
- 3.3 SQL データベースのテーブル構築
- 3.4 SQL データベースへの、SQL Management Studio からの I/O 確認
- 3.5 SQL データベースへの、API による I/O 確認

第4章 バッテリー記録アプリを、Azure サーバー上に公開する

- 4.1 Azure SQL のリソース作成と動作確認

- 4.2 Azure SQL データベースのテーブル構築と動作確認
- 4.3 クライアントからのコマンドラインベースで API 利用
- 第 5 章 起床と就寝を記録する Web ブラウザアプリを公開
- 5.1 起床と就寝のログを記録するアプリを設計
- 5.2 データベースの設計と SQLite データベースという選択
- 5.3 サーバー側: SQLite データベースへのアクセスを覚える
- 5.4 簡単なユーザー登録と認証を作成して、SQL へのアクセス I/F を組み上げる
- 5.5 Express フレームワークで簡単に実装
- 5.6 Azure Web App へ配置と設定の仕方
- 5.7 Vue.js フレームワークでクライアント側の UI を作成
- 5.8 クライアント側 UI を含めて Azure 上で動作確認をする
- 付録 A バッテリー残量記録の仕様
- 付録 B SQLite データベースをグラフィカルに参照する方法
- 付録 C Windows 向けに Electron でネイティブアプリ化する

<< 著者紹介 >>

窓川 ほしき

大学時代に、趣味で Windows アプリケーションの作成を始める。アプリは Vector で公開し、ダウンロードのカテゴリで人気 1 位を獲得。2016 年に Node.js と出会い「こんなに簡単にサーバーサイドのコードも書けるのか!」と感動、Web ブラウザベースのツール作成を開始する。「JavaScript での作成の手軽さと Azure での公開の簡単さを伝えたい」と、技術系同人誌の即売会イベントにて同人誌を頒布していたところ、商業出版の声がかかる。Web 上での名前は「\しまど」。最近のマイブームは劇場版 BLAME!。

<< 販売ストア >>

電子書籍:

Amazon Kindle ストア、楽天 kobo イブックスストア、Apple iBookstore、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」「モバイルサービス」を主要テーマに専門性の高いコンテンツ+サービスを提供するメディア事業を展開しています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

〒101-0051 東京都千代田区神田神保町 1-105

TEL 03-6837-4820

電子メール: np-info@impress.co.jp