報道各位

2019 年 8 月 27 日
株式会社インプレスR&D
https://nextpublishing.jp/

機械学習だけじゃない！　Python でサーバーレスアプリ開発！

『ほぼ Python だけでサーバーレスアプリをつくろう』発行

技術の泉シリーズ、8 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、『ほぼ Python だけでサーバーレスアプリをつくろう』(著者:長谷場 潤也、安田 譲)を発行いたします。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『ほぼPythonだけでサーバーレスアプリをつくろう』
https://nextpublishing.jp/isbn/9784844398974

著者:長谷場 潤也、安田 譲
小売希望価格:電子書籍版　1800 円(税別)／印刷書籍版　2000 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 164 ページ
ISBN:978-4-8443-9897-4
発行:インプレス R&D

＜＜発行主旨・内容紹介＞＞

本書は、「ほぼ」Python だけで AWS を利用したサーバーレスアプリケーションを構築するためのガイドブックです。

バックエンドに Chalice、フロントエンドに Transcrypt、ユニットテストと API テストに pytest、UI テストに Selene を利用します。

実装部分をアプリエンジニアが、テスト部分を QA エンジニアが、それぞれの専門を活かして解説しています。

(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。)

開発環境の構築から丁寧に解説

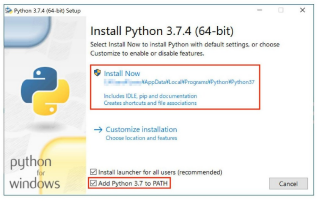
インストールしてください。

2.2.3 Windowsの場合

https://www.python.org/downloads/windows/からインストーラーをダウンロードします。「Python 3.7.4 - July 8, 2019」のリストから、ご利用の環境に合わせたファイルを選んでください。

ダウンロードしたexeファイルを開くとインストーラーが起動します。「Add Python 3.7 to PATH」にチェックを入れ、「Install now」をクリックしてください（図2.1）。以降はインストーラーの指示にしたがってインストールを完了させてください。

図2.1: Python3.7.4のインストーラー



インストールが完了したら、最後に環境変数を編集しましょう。ユーザー環境変数の`Path`に`%APPDATA%\Python\Python37\Scripts`を追加してください[7]。

2.3 Javaをインストールする

何度も繰り返しますが、本書のタイトルは「**ほぼ**Pythonだけでサーバーレスアプリをつくろう」です。残念ながら次のような場面ではJavaの力を借りなければなりません。

・DynamoDB Localを起動する（第4章）
・Transcryptでトランスパイルしたコードをminifyする（第5章）

とはいえ、ここで要求されているのは実行環境としてのJava Virtual Machineであり、プログラミング言語としてのJavaではありません。本書の範囲内でJavaのコードを書くことはありませんのでご安心ください。

なお、本書では**AdoptOpenJDK**を例にインストール方法を説明しますが、別の団体がビルドしたJDKを利用してもまったく問題はありません。

それではインストール作業に戻りましょう。https://adoptopenjdk.net/からインストーラーをダウンロードします。「Choose a Version」から「OpenJDK 11 (LTS)」を選択し、「Latest release」のリンクからダウンロードしてください。

ダウンロードしたファイルを開くとインストーラーが起動します。以降はインストーラーの指示にしたがってインストールを完了させてください。

> **Java is Still Free.**
>
> 「Javaは有償化されるんじゃなかったの？　インストールしても大丈夫なの？」と不安になった方もいらっしゃるでしょう。でも、ご安心ください。Javaは今後も無償で利用できます。
>
> リリースモデルがこれまでとは大きく変わったため、情報が錯綜するのは仕方がない面もあります。その一方で、意図的に誤った解釈をしているように見受けられる例もあるのが困りものです。Oracle嫌いをこじらせたアンチがOracleを叩くためだったり、SEO対策だけは完璧な自称技術ブログがPVを稼ぐためだったり……。
>
> このような誤った情報のせいで誤解されていた方は「**『Java 有償化』**で誤解する人になるべく分かりやすく説明するためのまとめ」[8]をぜひご一読ください。「ほぼPythonだけ」を銘打った本でこんなことを力説するのもおかしな話ですが、筆者たちはこういった正しい情報が広まってくれることを心より願っています。
>
> 8.https://togetter.com/li/1343743

2.4 Gitをインストールする

第6章でCI/CD環境を構築するために、分散型バージョン管理システムの**Git**を利用します。普段Gitをお使いでない方は、手順にしたがってインストールしてください。

2.4.1 Macの場合

http://git-scm.com/download/macからインストーラーをダウンロードします。

ダウンロードしたdmgファイル内のpkgファイルを開くとインストーラーが起動します。以降はインストーラーの指示にしたがってインストールを完了させてください。

2.4.2 Linuxの場合

`apt-get`、`yum`、`dnf`など、ディストリビューションに応じたパッケージ管理コマンドで`git`をインストールしてください。

2.4.3 Windowsの場合

https://git-scm.com/download/winからインストーラーをダウンロードします。

ダウンロードしたexeファイルを開くとインストーラーが起動します。以降はインストーラーの指示にしたがってインストールを完了させてください。

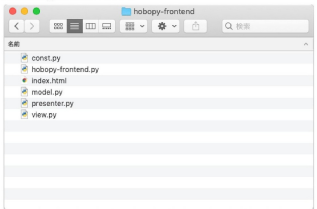
Python製のトランスパイラである Transcrypt で画面を実装

第5章　Transcryptで画面の実装をしよう

5.1 HTMLで画面を作成する

前章までの実装でバックエンドのWeb APIがひとまず完成しました。引き続き、本章ではフロントエンドを実装します。`hobopy`の下に`hobopy-frontend`というディレクトリーを作成し、その中で6つのファイルを実装していきましょう。

図5.1: hobopy-frontendディレクトリー



Webアプリのフロントエンドを実装するためには、HTML、CSS、JavaScriptが必要になります。本書ではデザインに**Bootstrap**[1]を利用し、独自のCSSを書くことはありません。また、スクリプトに関しては、Transcryptを通してPythonのコードをJavaScriptに変換します。つまり、JavaScriptを直接書くこともありません。

しかし、HTMLだけはそのまま扱わざるを得ません。しつこいようですが、本書のタイトルは「**ほぼ**Pythonだけでサーバーレスアプリをつくろう」なのです。観念してHTMLを作ってしまいましょう。

リスト5.1: hobopy-frontend/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>ほぼPythonだけでサーバーレスアプリをつくろう</title>
```

1.http://getbootstrap.com/

```
    <!-- ① Bootstrapを利用する -->
    <link rel="stylesheet"
      href="http://cdn.honokak.osaka/honoka/4.3.1/css/bootstrap.min.css">

  </head>
  <body>
    <nav class="navbar navbar-dark bg-dark">
      <a class="navbar-brand" href="#">
        ほぼPythonだけでサーバーレスアプリをつくろう</a>
      <button class="btn btn-outline-warning" id="new-todo" data-toggle="modal"
        data-target="#input-form">新規登録</button>
    </nav>
    <table class="table table-hover">
      <thead>
        <th scope="col"></th>
        <th scope="col">タイトル</th>
        <th scope="col">メモ</th>
        <th scope="col">重要度</th>
        <th scope="col"></th>
        <th scope="col"></th>
      </thead>

      <!-- ② ToDoリストの表示領域 -->
      <tbody id="todo-list">
      </tbody>

    </table>

    <!-- ③ 新規登録／変更用モーダルダイアログ領域 -->
    <div class="modal fade" id="input-form" tabindex="-1" role="dialog">
      <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title" id="modal-title"></h5>
            <button type="button" class="close" data-dismiss="modal"
              aria-label="Close">
              <span aria-hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
```

ユニットテスト、API テストの方法も解説

第8章　pytestでユニットテストをしよう

この章では、テストレベルでいえば実装の直後に当たる**ユニットテスト**の自動化に触れていきます。関数やメソッドなど、テストレベルの中では一番小さい単位で実施するテストです。

8.1　Pythonにおける xUnit

前章で説明したV字モデルのなかで、「品質を確認する工程」の一番最初に描かれていたものがユニットテストです。このユニットテストですが、以前よりテストコードを書いて自動化することが行われています。

よく使われるのが**xUnit**と呼ばれるユニットテスト用のフレームワークです。QAの用語では**テストハーネス**と呼ばれることもあります。Javaであれば「JUnit」、PHPであれば「PHPUnit」と、それぞれの言語で同様の役割や動作をするフレームワークが存在します。

そして、Pythonでそれに当たるものは何かというと……「PyUnit」ではなく**unittest**[1]という標準ライブラリになります。また、unittestを強化したライブラリもあり、現在一般的によく使われるのが**pytest**[2]になります。この章では主にpytestを使って、テストを実施します。

よかった……下手したらこの章のタイトルが「unittest（ライブラリ名）でユニットテスト（実施するテストレベル名）をしよう」になって、大変ややこしいことになるところでした。

8.2　ユニットテストの環境を用意する

では、Pythonでのユニットテストを作成していきましょう。まずはpytestのインストールからですが、他のライブラリ同様にpipコマンドでインストールができます。なお、今回のユニットテストはhobopy-backendの中のロジックに対して行いますので、インストールする仮想環境もhobopy-backendになります。

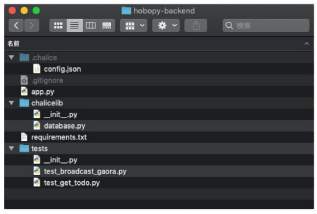
```
(hobopy-backend) $ pip install pytest
```

また、ユニットテストのコードですが、hobopy-backendの直下にtestsというディレクトリーを作成し、そこに作っていきます。なお、ここで作成したディレクトリーは第3章で触れられたように、chalice deployコマンドでのデプロイの対象にはなりません。

本章でこのあと作成するテストコードのファイルも含めると、このような構成になります。

1.https://docs.python.jp/3/library/unittest.html
2.https://docs.pytest.org/en/latest/index.html

図8.1: テストコードを含めた hobopy-backend ディレクトリー



8.3　ユニットテストを書いてみる

インストールができましたので、ここからテストコードを書いていきます。

どのプログラムに対してのテストコードがよいかを考えたのですが……どうもこれまで作ってきたToDoアプリのにはこのような関数がありました。

リスト8.1: hobopy-backend/chalicelib/database.py

```
def broadcast_gaora(memo):
    if len(memo) > 30:
        memo = "ホームラン" + memo
    if "ホームラン" in memo:
        memo = memo.replace("ホームラン", "「イッツ！」")
        memo += "「ゴーヌ！」"
    return memo
```

メモの文字数が30を超えていたら、頭に「ホームラン」という文字列を加える。さらに「ホームラン」と書かれていた場合は、その文字を「「イッツ！」」に変更し、「「ゴーヌ!」」という文字列を語尾に加える関数です。これはどうやらスポーツ専門チャンネルGAORA SPORTS[3]のプロ野球中継で、ホームランが飛び出した時の近藤アナウンサーの実況を再現する……という関数のようですが……。

なんてとってつけたような単体テストのしやすい関数なのでしょう！　せっかくなのでこちらでテストを実施していきます[4]。

まずは、ごく単純なテストコードを作ってみました。

3:https://www.gaora.co.jp
4.えっと……ほんとはこんな仕様ないです。ちょうど良さそうなのをでっち上げたかったのですごめんなさいごめんなさい。

＜＜目次＞＞

＜＜著者紹介＞＞

長谷場 潤也（はせば じゅんや）
技術系同人サークル「Thunder Claw」のニンジャスレイヤー好きのほう。第1章から第6章までを担当。埼玉西武ライオンズとクイズマジックアカデミーをこよなく愛するWebエンジニア兼Androidエンジニア。好きな大阪桐蔭は中村剛也、好きな富士大学は外崎修汰、好きなクイズ形式はアニメ＆ゲーム並べ替え。

安田 譲（やすた じょう）
技術系同人サークル「Thunder Claw」のポプテピピック好きのほう。第7章から第12章までを担当。根っからの特撮好

きな元システムエンジニア、現 QA エンジニア。雨宮慶太がデザインする怪人や、坂本浩一がこだわる足のカットには日々感銘を受けている。過去 6 つの特撮作品にてエキストラ出演経験あり。

＜＜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。

【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イア&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