

2019年8月9日

株式会社インプレスR&D

<https://nextpublishing.jp/>

プログラミングの人気著者 田中賢一郎氏のテキスト、好評につき続編発行！

『Web 技術速習テキスト 実践編』

Web アプリを作ってみよう。

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、『Web 技術速習テキスト 実践編』(著者:田中 賢一郎)を発行いたします。

『Web技術速習テキスト 実践編』

<https://nextpublishing.jp/isbn/9784844378174>



著者:田中 賢一郎

小売希望価格:電子書籍版 1800円(税別)／印刷書籍版 2200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／モノクロ／本文 252ページ

ISBN:978-4-8443-7817-4

発行:インプレス R&D

<< 発行主旨・内容紹介 >>

本書は、プログラミングの人気著者、田中賢一郎氏が執筆し、2019年7月にインプレスR&Dより発行した『Web 技術速習テキスト』の続編です。前著がとても好評なので、実際にWebアプリを作成する「実践編」を急遽発行いたしました。

Web関連技術の進歩は目覚ましいものがあります。クラウドファーストという言葉があるように、クラウド前提のシステム開発が一般的になってきました。本書では、前著の『Web 技術速習テキスト』を読み終えた方を対象に、Vue.js や Vuetify などのフレームワーク、Firebase などを使って、Webアプリを作って実際にクラウド上で動かしてみます。

多くのアプリ例に触れることで、実際に使えるスキルの習得を目指します。

(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。)

1.2 プロジェクトを作って公開する

1.2 プロジェクトを作って公開する

プロジェクトをクラウドに公開するところからでもアクセス可能になります。スマホなどで閲覧すると自分の作ったプロジェクトがインターネット上に公開されていることを実感できると思います。

まず、CLIでVueのプロジェクトを作ります。今回はc:\WebAppというフォルダの下でmy-first-appというプロジェクトを作ったという前提で説明をします。別のフォルダ・別のプロジェクト名を使用する場合は適宜置き換えてください。以下のようにvueコマンドを使ってプロジェクトを作成します。

vue create プロジェクト名

途中で質問される場合は、単に`Enter`キーを押下してデフォルト設定を進めてください。

```
c:\WebApp>vue create my-first-app
Vue CLI v3.5.1
? Please pick a preset: default (babel, eslint)
Vue CLI v3.5.1
✔ Creating project in c:\WebApp\my-first-app.
```

プロジェクトの作成が終わったら、my-first-appというフォルダが作成されているので、cdコマンドを使ってそのフォルダへ移動し、npm run serveコマンドを実行します。

```
c:\WebApp>cd my-first-app
c:\WebApp\my-first-app>npm run serve
```

しばらくすると以下のような画面が表示されます。



途中で以下のようなウィンドウが表示された場合 (Windowsの場合)、アクセス許可するボタンをクリックしてください。



ブラウザを起動して、`http://localhost:8080`にアクセスしてください。



4.1 セットアップ

4.1 セットアップ

世界標準のプロジェクトではCDNへのリンクを挿入してBootstrapを使用しました。今回はVue用に特化したbootstrap-vueを使用します。Bootstrapで使用するclassを使ってスタイルを指定する方法だけでなく、Bootstrap用の専用タグが使用可能になります。

4.1.1 プロジェクトの作成

BootstrapとVueを組み合わせて使うのであれば、Bootstrap-vueが便利です。

<https://bootstrap-vue.js.org/>
以下のURLを参考にプロジェクトを作成します。

<https://bootstrap-vue.js.org/docs>

今回はVue CLI3を使いますが、バージョンが更新されるとプロジェクトの作成手順も変更される可能性が高いため、最新の情報を確認してください。→はvueのバージョンを確認するコマンドオプションです。

vue -V

```
c:\WebApp>vue -V
Vue CLI v3.5.1
```

vue create プロジェクト名でプロジェクトの作成を開始します。今回はbv-basicというプロジェクトを作成します。

```
c:\WebApp>vue create bv-basic
Vue CLI v3.5.1
? Please pick a preset: (Use arrow keys)
> default (babel, eslint)
```

プリセットの選択では、デフォルトのまま`Enter`キーを押します。プロジェクト作成が完了したら、cd bv-basicでプロジェクトのフォルダにカレントディレクトリを移動して、npm run serve

コマンドでプロジェクトを開始します。ブラウザで`http://localhost:8080`にアクセスしてサーバが稼働しているか確認してください。

4.1.2 bootstrap-vueのインストール

別のターミナルを開き、npmコマンドでbootstrap-vueをインストールします。

npm i bootstrap-vue

```
c:\WebApp>npm i bootstrap-vue
```

以下のようにmain.jsファイルに4行を追加します。

```
import Vue from 'vue'
import App from './App.vue'

import BootstrapVue from 'bootstrap-vue' //ここから4行を追加
import 'bootstrap/dist/css/bootstrap.css'
import 'bootstrap-vue/dist/bootstrap-vue.css'
Vue.use(BootstrapVue)

Vue.config.productionTip = false

new Vue({
  render: h => h(App),
}).$mount('#app')
```

これでBootstrap-Vueが使用できるようになります。本家のBootstrapではclass属性を使用してきたがBootstrap-Vueでは、classを使うこともできますが、専用のタグが用意されています。helloWorld.vueファイルの内容をすべて削除して、以下のように修正してください。

```
<template>
  <div>
    <div class="jumbotron">
      <h1>Hello, World!</h1>
```

6.1 Vuetify

6.1 Vuetify

Vuetify は「マテリアルデザイン」と呼ばれるデザイン手法に沿ったページを作成するためのフレームワークです。他の多くの CSS フレームワークもそうですが、Vuetify も Bootstrap の影響を受けているため、Bootstrapを知っていると学習が容易になります。

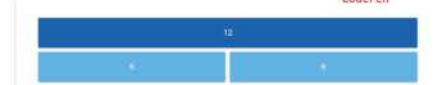
Vuetify はとてもリッチなフレームワークです。ここでは、公式ページのドキュメントを読み進めながら自力でいろいろ試せるようになることを目標とします。基本的な事項について説明します。公式サイトには数多くのサンプルが掲載されています。サンプルのソースコードを見たり、CodePen で試したりすると理解が深められます。例えば、グリッドの説明ページを見ましょう。

<https://vuetifyjs.com/ja/framework/grid>

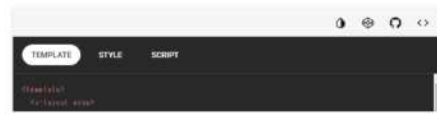
Usage

The `v-container` can be used for a center focused page, or given the `flex` prop to extend to full width. `v-row` is used for repeating sections and contains the `v-col`. The structure of your layout will be as follows: `v-container -> v-layout -> v-flex`. Each part of the grid chain is a flexbox element. The final `v-col` automatically sets its children to have `flex: 1`.

For convenience reasons, these components automatically transform attributes into classes, allowing you to write `no-hover="true" @click="doIt" instead of no-hover="true" @click="doIt" class="no-hover". The only exception are data attributes as they are left as-is.`



それぞれのサンプルの右上にはテーマの変更、CodePen、GitHub のページ、ソースコードと 4 つのアイコンがあります。ソースコードボタンをクリックすると TEMPLATE、STYLE、SCRIPT と必要に応じてタブが表示され、そのサンプルを作成するためのソースコードを見ることができます。



Vuetify と Bootstrap をある程度知っていれば、サンプルを見るだけで Vuetify の使い方の予想がつくでしょう。CodePen と連携しているため実際に変更を加えてみると、その動きがよくわかります。HTML、CSS でできているとは思えないようなカッコいい部品もたくさんあります。ぜひいろいろと試してみてください。

6.1.1 セットアップ

手を動かしながら確認するのが一番です。以下の URL に沿ってプロジェクトを作成します。

<https://vuetifyjs.com/ja/getting-started/quick-start>
プロジェクト名は `vuetify-basics` としました。



プロジェクトが作成されたら、そのディレクトリに移動して `vue add vuetify` コマンドを実行します。



プリセット選択画面では Default を選択します。



`npm run serve` コマンドで実行し、ブラウザから <http://localhost:8080> にアクセスします。以下のような画面が表示されます。

7.1 サインアップ&ログイン

7.1 サインアップ&ログイン

ユーザーを新規登録するサインアップ、登録したユーザーを認証するログイン、この2つは多くのサービスで基本となる機能です。Firebase でこれらを実装するか手を動かしながら見ていきましょう。

7.1.1 プロジェクトの作成

Vue.js のプロジェクトを `firebase-auth` という名前で作成します。設定等の質問がある場合は **Enter** キーを押下してデフォルトを選択し、進んで下さい。

```
vue create firebase-auth
```



プロジェクトが作成されたら、そのディレクトリに移動して `vue add vuetify` コマンドを実行して `vuetify` モジュールを追加します。



以下のプリセット選択画面では Default を選択します。



`src/App.vue` ファイルを以下のように修正します。

```
● src/App.vue
<template>
  <div class="app">
    <h1>firebase-auth</h1>
    <div class="container">
      <div class="row">
        <div class="col">
          <div class="card">
            <div class="card-content">
              <div class="text">
                <h2>Sign Up</h2>
                <div class="input-field">
                  <input type="text" v-model="email" />
                </div>
                <div class="input-field">
                  <input type="password" v-model="password" />
                </div>
                <div class="input-field">
                  <input type="text" v-model="name" />
                </div>
                <div class="input-field">
                  <input type="text" v-model="password2" />
                </div>
                <div class="input-field">
                  <input type="checkbox" v-model="terms" />
                </div>
                <div class="input-field">
                  <input type="button" value="Sign Up" />
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</template>
<script>
  export default {
    data() {
      return {
        email: '',
        password: '',
        name: '',
        password2: '',
        terms: false
      }
    }
  }
</script>
```

8.1 ローカルアプリの作成

8.1 ローカルアプリの作成

これから Vuetify と Firebase を組み合わせてアプリを作成していきます。最初から完成形を目指すのではなく、まずユーザーインターフェースのみを構築し、必要な機能や使い勝手などを検証します。

8.1.1 アプリの概要

アプリを使用するシーンとして、講師が数人、生徒が数十人といった小規模な教室を想定します。生徒の学習状況を講師間で共有するためのアプリを作ります。以下のような要件を満たすものとしていきます。

- ・生徒はログイン不要（誰かが匿名でログインしておく）、生徒は毎回のレッスン後にページにアクセスし、自分の名前を選んでコメントを記入
- ・講師としてログインすると、生徒を認べるようになり、過去の生徒のコメント、他の講師のメモが閲覧できる。講師は生徒のコメントに対してメモを追記できる。
- ・各種情報は Firebase の Firestore に保存
- ・認証は Firebase の Authentication を使用

講師のアクションフロー	講師のアクションフロー	講師のアクションフロー
ログイン画面	1. 日付選択 2. 生徒名選択 3. コメント入力	1. 生徒名選択 2. メモ入力
受講画面	生徒用画面（匿名ログイン）	講師用画面

8.1.2 プロジェクトの作成

プロジェクト名は class-tracker としました。Vue のプロジェクトを作成し、Vuetify を追加してください。選択が促された場合はデフォルトの選択肢を選んでください。

```
vue create class-tracker
cd class-tracker
vue add vuetify
```

8.1.3 UI (ユーザーインターフェース) の構築

まずはUIを作成し、デモデータを使ってローカルで動かしてみましょう。デスト用モックアップなので、生徒用画面と講師用画面の両方が表示されるものとします。上段が生徒用画面、下段が講師用画面です。



生徒の入力には Steppers コンポーネントを使用しました。これは段階的にユーザーを誘導するのに適したコンポーネントです。日付、氏名、コメントと順に入力を使います。講師用のデータ入力には Data tables コンポーネントを使用しました。

1つのページにいろいろな要素を詰め込みすぎると可読性が低下するので、今回は以下のように3つのコンポーネントを組み合わせました。



```
graph TD
    App[App.vue] --- Student[StudentView.vue]
    App --- Teacher[TeacherView.vue]
```

<<目次>>

第1章 Vue cli を使ってみる

第2章 Vue.js の基礎

第3章 コンポーネント

第4章 アルバム(bootstrap + Vue)

第5章 ブログ(contentful + Vue)

第6章 スケジューラー(Vuetify)

第7章 ログイン(Firebase 認証)

第8章 授業記録アプリ

<<著者紹介>>

田中 賢一郎

慶應義塾大学理工学部修了。キヤノン株式会社に入社し、デジタル放送局の起ち上げに従事。その間に単独でデータ放送ブラウザを実装し、マイクロソフト(U.S.)ヘソースライセンスし、Media Center TV チームの開発者としてマイクロソフトへ。マイクロソフトではWindows、Xbox、Office 365などの開発・マネージ・サポートに携わる。2016年に中小企業診断士登録後、セカンドキャリアはIT教育に携わると決め、2017年春にFuture Codersを設立。「プログラミング教育を通して一人ひとりの可能性をひろげる」という理念のもと、楽しいだけで終わらない実践的な教育を目指している。

<<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 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