報道各位

2019 年 6 月 26 日
株式会社インプレスR&D
https://nextpublishing.jp/

Vue.js の設計に自信がないエンジニアのための指南書！

『後悔しないための Vue コンポーネント設計』発行

技術の泉シリーズ、6 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、『後悔しないための Vue コンポーネント設計』(著者:中島 直博)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。

『後悔しないためのVueコンポーネント設計』
https://nextpublishing.jp/isbn/9784844398691

著者:中島 直博
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 64 ページ
ISBN:978-4- 8443-9869-1
発行:インプレス R&D

＜＜発行主旨・内容紹介＞＞

本書は、Vue.js を利用してシングルページアプリケーションの作成を考えているエンジニア、特にコンポーネントの設計や分類に悩んでいる方のためのガイドブックです。

テストの書き方がわからない、コンポーネントのアンチパターンを知りたい、といった方のために、テストしやすい、またはしづらいコンポーネントとはなにか、単体テストの書き方などについて丁寧に解説しています。

〈本書の対象読者〉

・Vue.js のコンポーネント設計に自身がない

・Vue.js のコンポーネントの分類で悩んでいる

・Vue.js のコンポーネントのアンチパターンを知りたい

・Vue.js のコンポーネントのテストの書き方がわからない

（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。）

著者の経験を踏まえ、テストしやすいコンポーネントについて紹介

第2章　テストしやすいコンポーネントと、テストしづらいコンポーネント

この章では、どういったコンポーネントがテストしやすく、どういったコンポーネントがテストしづらいのかについて、筆者の経験を踏まえた考えを紹介します。

2.1　テストしやすい/しづらいコンポーネント

筆者が考えるテストしやすいコンポーネントとは、次のようなものです。

・機能が少ない

・依存が少ない/少ない

・状態をもたない

また、これらの項目のひとつでも逆をいくコンポーネントは、テストしづらいといえます。

これらの項目が、どうテストのしやすさと結びつくのかについて解説します。

2.2　機能を少なくシンプルに保つ

コンポーネントの機能が少なければ少ないほど、テストはしやすくなります。ここでの**機能**とは、methodsの数ではなく、コンポーネントが担う役割のことです。

propsで受け取った値を表示するだけでなく、dataで自身の状態を操作したり、レンダリングのためにcomputedで複雑な計算をいくつもしたりといったコンポーネントは**機能**が多く、テストしづらいといえます。

もしコンポーネントの機能が多いなと感じたら、それはコンポーネントの役割がうまく分割できていないサインかもしれません。役割を見極めて、ちょうどいい粒度でコンポーネントを分割できると、アプリケーションの構造としても、テストのためにもよいのです。

2.3　依存は少なく

コンポーネントが依存しているものが少なければ少ないほど、テストはしやすくなります。

依存の数は、そのコンポーネントの再利用性にも関わってきます。たとえば、VuexのStoreやRouterにべったり依存したようなコンポーネントは、他の場所や用途で使うのは難しいでしょう。

リスト2.1: 依存の多いコンポーネント

```
<template>
  <div>
    <div
      v-for="item in items"
      :key="item.id"
    >
      <span>{{item.label}}</span>
      <button @click="clickItem(item)">
        {{item.label}}
      </button>
    </div>
  </div>
</template>

<script>
import { mapState } from "vuex"
export default {
  name: "BigComponent",
  computed: {
    ...mapState(["items"]),
  },
  methods: {
    clickItem(item) {
      this.$router.push({
        name: "itemDetail",
        params: {
          name: item.name,
          id: item.id,
        },
      })
    },
  },
}
</script>
```

依存を少なくするには、コンポーネントの役割を分割することが大事になります。たとえばリスト2.1の例では、Storeから値を取り出すコンポーネントと値を表示するコンポーネントに分けることで、値を表示するコンポーネントは他の場所でも使えるようになります。

リスト2.2: Storeから値を取り出すコンポーネント: Container.vue

```
<template>
  <div>
    <Item
      v-for="item in items"
      :key="item.id"
```


テスト環境の構築、テストの書き方を解説

第6章　テスト実行環境の構築

本章ではテスト環境の構築と、テストの書き方について解説します。

まず、Vueコンポーネントのテストを実行するための環境を構築していきましょう。

6.1　Vue CLIを使った環境構築

これから新しいプロジェクトを作成するのであれば、Vue CLI v3[1]を使った環境構築を強く推奨します。Vue CLIを使ったテスト環境の構築はとても簡単です。まだテストを書いたことがない方でも、Vue CLIを使えばテスト実行環境の構築とテストの実行がすぐに行えます。

6.1.1　Vue CLIでプロジェクトを作成する

次のコマンドは、Vue CLIによるプロジェクトの新規作成を行うものです。

```
$ npx vue create my-project
```

このコマンドを実行すると、次のように対話形式でプリセットの選択を求められます。

```
Vue CLI v3.0.1
? Please pick a preset: (Use arrow keys)
> ts (vue-router, vuex, sass, babel, typescript, pwa, eslint, unit-jest)
  js (vue-router, vuex, sass, babel, pwa, eslint, unit-jest)
  default (babel, eslint)
  Manually select features
```

tsとjsプリセットにあるunit-jestという項目に、Jestというテストランナーを使った環境構築を含んでいます。このふたつのどちらかを選択すると、テスト環境を含む新規プロジェクトがすぐに作成されます。

プリセットではなく自分で使用したいものを選ぶ場合は、**Manually select features**を選択して、次に表示される**Unit Testing**を選択します。

```
Vue CLI v3.0.1
? Please pick a preset: Manually select features
```

1 https://cli.vuejs.org/

```
? Check the features needed for your project:
 ◉ Babel
 ◯ TypeScript
 ◯ Progressive Web App (PWA) Support
 ◯ Router
 ◯ Vuex
 ◯ CSS Pre-processors
 ◉ Linter / Formatter
>◉ Unit Testing
 ◯ E2E Testing
```

項目を進んでいくと、テストに使うテストランナーの選択が表示されるので、**Jest**を選択します。

```
? Pick a unit testing solution:
  Mocha + Chai
> Jest
```

次の選択として各種設定を**package.json**に書くか、それぞれをファイルとして出力するかが表示されます。

```
? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? (Use arrow keys)
> In dedicated config files
  In package.json
```

Jestの設定は編集する頻度がそれほど高くないので、プロジェクト直下にファイルが増えることを避けたい方は**package.json**を選択するとよいかと思います。

6.1.2　テストの実行

Vue CLIで作成したプロジェクトの**package.json**には、scriptsに`test:unit`というコマンドが用意されているので、これを実行するとテストが実行できます。

実行すると次のようなテスト結果が出力されます。

```
$ npm run test:unit

> my-project@0.1.0 test:unit /path/to/test/my-project
> vue-cli-service test:unit

 PASS  tests/unit/HelloWorld.spec.js
  HelloWorld.vue
```

実際に Vue コンポーネントのテストの書き方を紹介

第7章　テストを書く

本章ではVueコンポーネントのテストの書き方について紹介します。テストを実行するテストランナーにはJestを、コンポーネントを操作するライブラリとしてvue-test-utilsを使います。

7.1　サンプルアプリケーション

テストを書いていく対象として、サンプルのアプリケーションを用意しました。

図7.1: top

機能としては、ヘッダーのコンポーネントで、/ページと/aboutページを切り替えるだけのシンプルなものになっています。
解説では主要なテストケースに限定しています。全てのテストケースはソースコードか、付録A「テストコード」を参照してください。

7.2　テストの実行方法

テストの実行は、次のコマンドで行います。

```
$ npm run test:unit
```

Vue CLIで作成したプロジェクトのテストでエラーが起きるときは

Vue CLI v3で作成したプロジェクトで、npm run test:unitを実行したときに、次のようなエラーが表示される場合があります。

リスト7.1: テスト実行時のエラー

```
SyntaxError: Unexpected string
    at ScriptTransformer._transformAndBuildScript
    (node_modules/jest-runtime/build/script_transformer.js:403:17)
```

その場合、jest.config.jsに次のようにcache: falseを加えてみてください。

リスト7.2: jest.config.js

```
module.exports = {
  // ...省略
  cache: false,
}
```

これでも直らない場合には、次のissueのコメントにある対策をいくつか試してみてください。
Default unit tests are failing -https://github.com/vuejs/vue-cli/issues/1879

7.3　ディレクトリとファイル構成

サンプルアプリケーションの/src配下の、ディレクトリ構造と、ファイル構成は次のようになっています。

```
./src
├── App.vue
├── assets
├── basics
|   ├── Logo.vue
|   └── SiteTitle.vue
├── components
|   ├── Menu
|   |   └── MenuItem.vue
|   └── Menu.vue
├── containers
|   └── GlobalHeader.vue
├── pages
|   ├── About.vue
|   └── Root.vue
```

36 | 第7章　テストを書く

第7章　テストを書く | 37

＜＜目次＞＞

＜＜著者紹介＞＞

中島 直博

株式会社ピクセルグリッド所属のフロントエンド・エンジニア。v0.10 から Vue.js を使い始め、業務/個人を問わず数多くのプロジェクトで採用経験あり。

＜＜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