

Vue.js 超入門 (概念的理解)

発表の経緯

現場で Vue.js を使っているのですが、
大きなプロジェクトの細かいところばかり見ている。

- 基本的な概念
- 全体がどうつながっているか

といったことをちゃんと理解したいと思った。

前提知識

- JavaScript
- Node.js
 - JavaScript の実行環境であるということ
- npm
 - Node.js のパッケージ管理ツールであるということ

概要

1. Vue.js とは何か
 - 現場でよく見る構成
2. Vue CLI とは何か
 - 全体のつながりを把握しよう
3. 定番ライブラリ
 - vue-router
 - vuex-store

Vue.js とは何か

- JavaScript のライブラリ
 - Vue オブジェクトを提供する。
- リアクティブ
 - Vue が画面要素とデータやメソッドを結びつける。
- プログレッシブ
 - 必要な機能だけ使えば良い。
 - フレームワークにもなる。

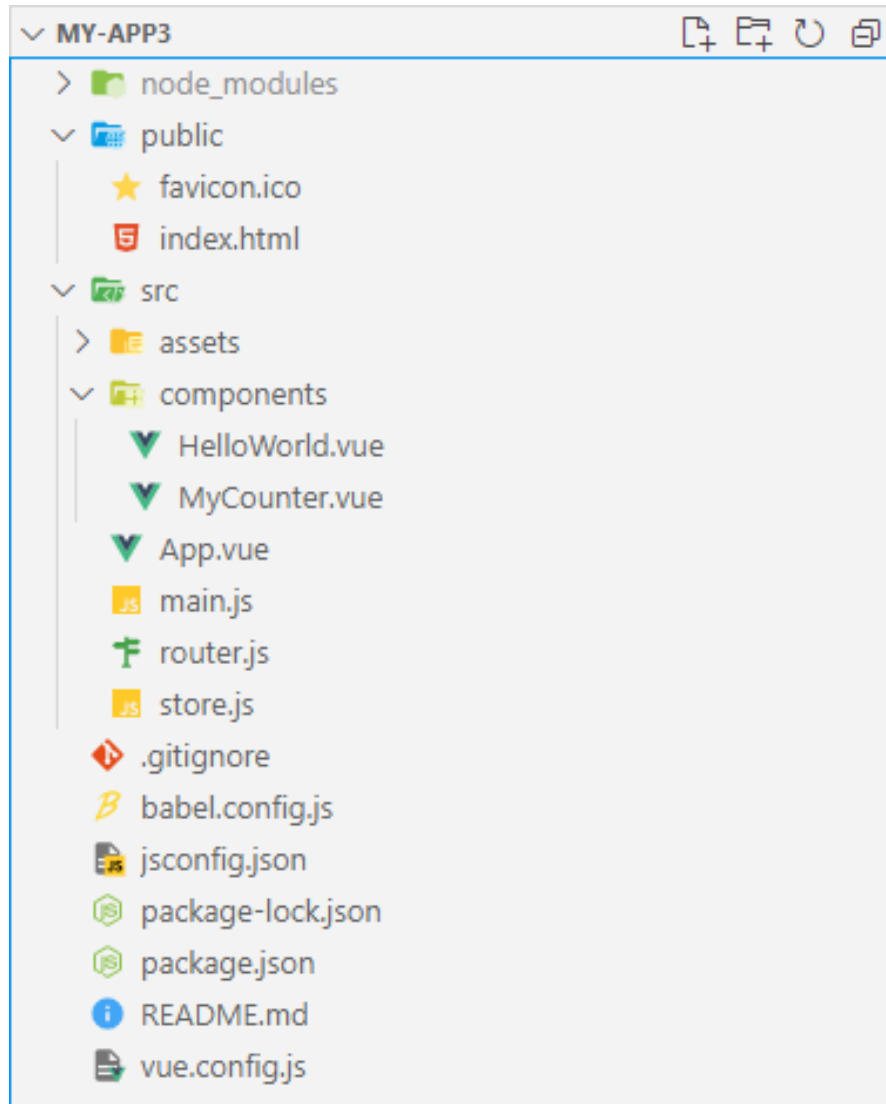
もっとも簡単な使い方

```
<script src="https://unpkg.com/vue@3.2.31"></script>
<div id="app">{{ msg }}</div>
<script>
  Vue.createApp({
    data() {return {msg: 'Hello, World.'}}
  }).mount('#app')
</script>
```

ライブラリを読み込んで Vue オブジェクトを作るだけの HTML。

- ここにメソッドとか追加して作り込める。
- UI -> data 方向のバインディングも可能。

現場でよく見る構成



プロジェクトをひな形から作成している。

html ではなく vue ファイルを書くようになる。

router や store といった定番ライブラリもインストールしている。

実際はもっとフォルダ分けすると思います。

プロジェクトをひな形から作成

例1 (主に Vue.js 2.x の場合)

```
> vue init webpack my-app
```

例2 (主に Vue.js 3.x の場合)

```
> vue create my-app3
```

- この "vue" は Vue.js ではなく vue-cli のコマンド。

Vue CLI とは何か

Vue CLI は、Vue.js での開発を補助するツール

- npm でインストール。
- コマンドは **vue**。

できること

- ひな形からプロジェクト作成
- vue ファイル/プロジェクトのビルド
- 開発用サーバの起動
- テストの実行
- etc.

ビルドは何をするか

以下のような vue ファイルを Web ブラウザで実行できる形にする。

```
<template>
  <div><p>{{ msg }}</p></div>
</template>

<script>
export default {
  name: 'HelloWorld',
  data() {return {msg: 'Hello, World.'}}
}
</script>
...
```

プロジェクトのビルド

ファイル	役割 (例)
public/index.html	main.js が組み込まれエントリポイントとなる。
src/main.js	App.vue をマウントする JS になる。
src/App.vue	router からの vue コンポーネントの受け皿。
src/router.js	パスや条件に応じて表示するコンポーネントを決める。
src/components/xx.vue	vue コンポーネント。

※ import/export によりこれらが連動する。

全体のつながりを把握しよう

1. 基本は import/export を追いかける。
2. コンポーネントの親子関係を知る。
 - データの受け渡しがどうなっているか等。
3. `$router`, `$store` といった約束事を覚える。
 - `router` を持つコンポーネントやその子孫は `this.$router` で `router` にアクセスできる。

その他、ビルドにより起こること

- 難読化
- Babel の実行
 - ES2015 の構文を古い構文に変換
- Scoped style の解決
- etc.

定番ライブラリ

- vue-router
 - 画面遷移をコントロール (SPA を構築) する
- vuex-store
 - 状態 (変数とか) 管理のためのデータフロー

main.js でマウントする vue に適用すれば、各コンポーネントから使えます。

```
const app = createApp(App)
app.use(router)
app.use(store)
app.mount('#app')
```

vue-router

router.js (かなり省略してます)

```
import { createRouter } from 'vue-router'
import HelloWorld from '@components/HelloWorld'

export default createRouter({
  routes: [
    { path: '/',
      name: 'HelloWorld',
      component: HelloWorld },
    ...
  ]
})
```

vuex-store

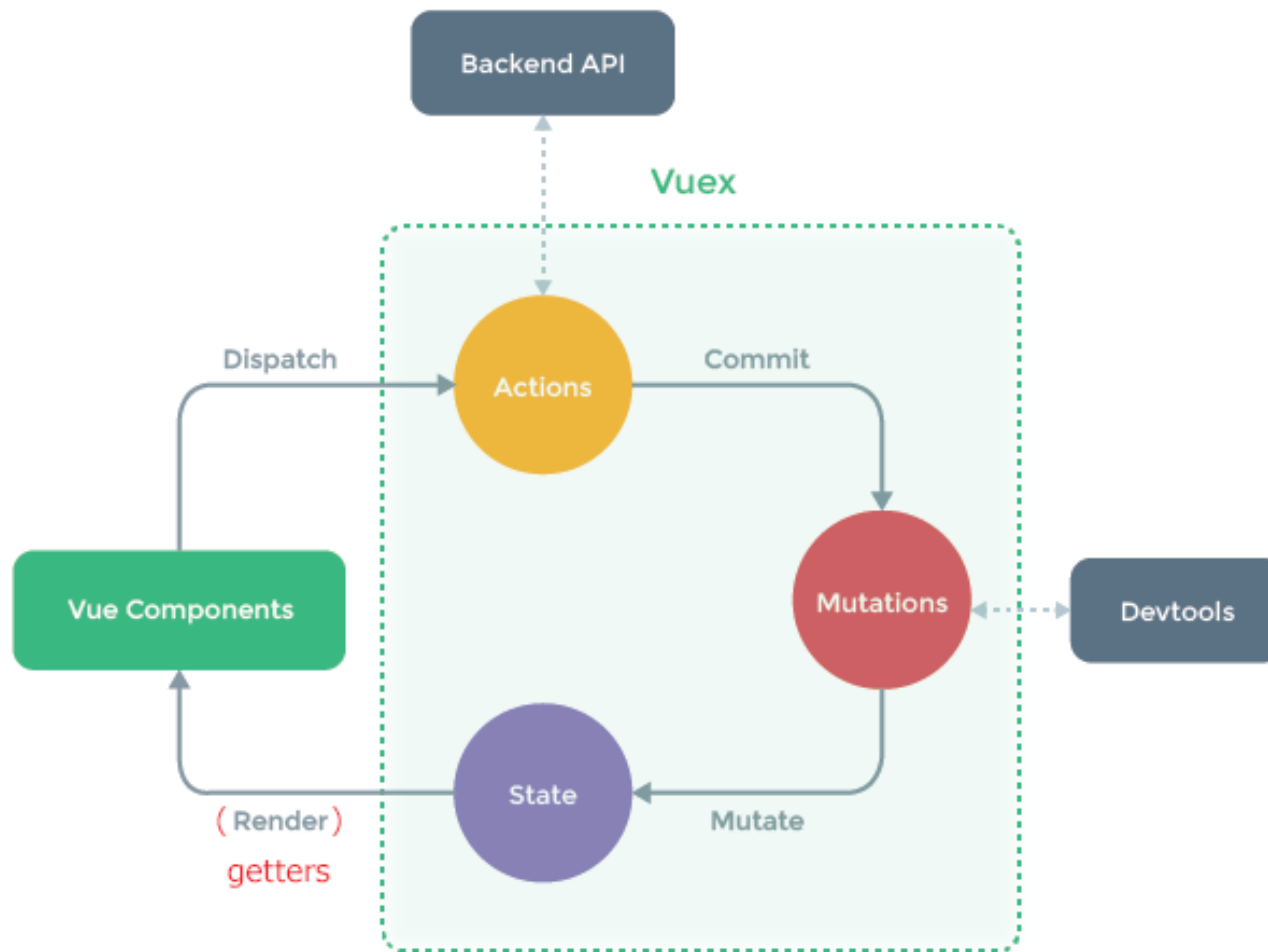
store.js (かなり省略してます)

```
import { createStore } from 'vuex'

export default createStore({
  state: {count: 0},
  getters: {
    count (state) {return state.count}},
  mutations: {
    increment (state) {state.count++}},
  actions: {
    increment (context) {context.commit('increment')}}
})
```

vuex-store によるデータフロー

「Vuex とは何か？」 (<https://vuex.vuejs.org/ja/>) より



まとめ

- 現場の手垢にまみれたコードを知りすぎる前に、
自分で一から作ったきれいなプロジェクトを見ておきましょう。
- ある機能を実装する方法はたくさんあります。
テクニックに走る前に、ツールのコンセプトを理解しましょう。