

エキスパートVueify

v-sapporo MeetUp #2

自己紹介

- kitaharata (Takuro Kitahara)
- kitaharata.github.io
- JavaScript Python Go PHP C# Ruby Java
- Angular(2)→Riot→Vue.js→(React)
- Vue.js 2年6ヶ月 Vuetify 1年9ヶ月 Nuxt.js 1年

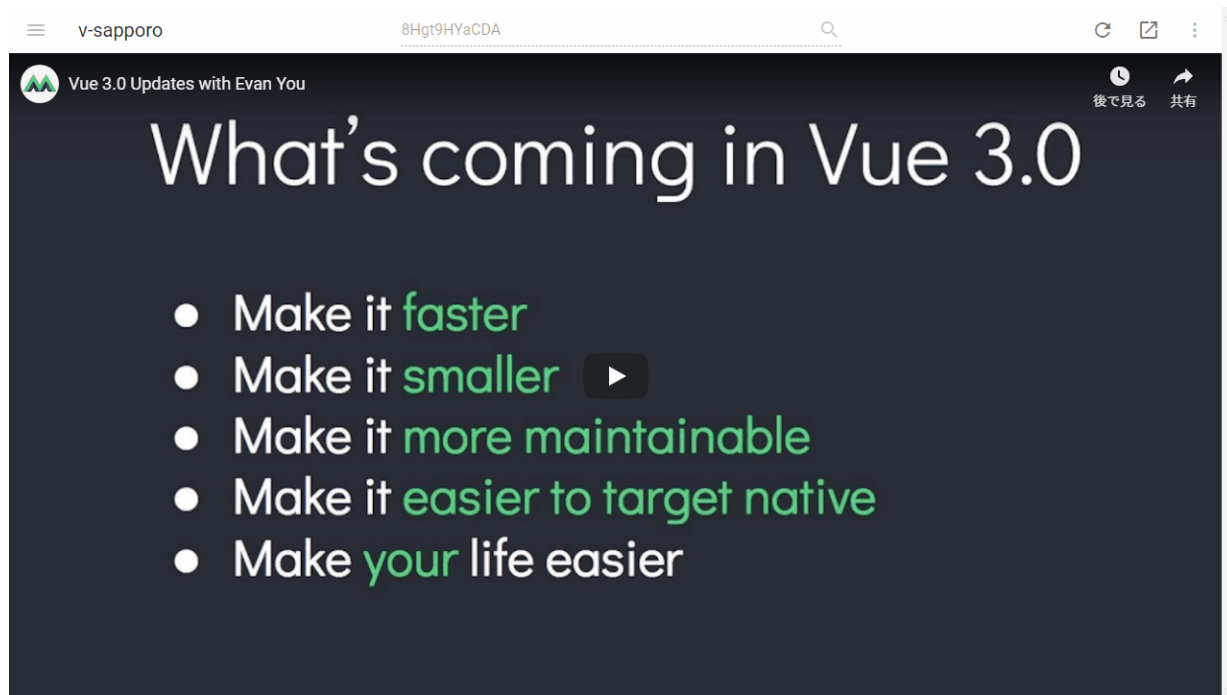
作品



Nijisanchi

- nijisanchi.netlify.com
- CDN + PWA
- Vue.js + Vue Router + Vuetify
- index.htmlのみ 125行
- 開発期間 3日

Vuetify使用例



Vuetifyとは

- Vue.jsで開発されたコンポーネントフレームワーク
- マテリアルデザインをベースとしており、BootstrapやSemantic UIのような柔軟なデザインを構築することができる
- v2はMD2に対応 (以降v2.0.14に関する解説)
- vuetifyjs.com

コンポーネントフレームワーク

- フレームワークによる制限でUIのモックアップやプロトタイプを作りやすく、ユーザーにとっても分かりやすい設計を実現することができる



Vuetifyのコンポーネント

- Vue.jsでおなじみのプロパティ、スロット、イベント、関数を最大限に生かしており、簡潔な記述でデザインをすることができる
- Vue.jsでおなじみのAPI、ディレクティブも問題なく使用することができるため、さらに簡潔な記述でデザインをすることができる

分かりやすいプロパティ

- 名前と型を統一することによって速く作れる
- Pugとの相性も良い

```
1 <v-icon color="#1da1f2">mdi-twitter</v-icon>
```

```
2 |
```

使いやすいプロパティ

- v-bind、三項演算子、動的な属性名に対応しているためテンプレート内だけで柔軟なケースに対応できる

```
1 <v-icon>{{ String ? 'mdi-star' : 'mdi-star-outline' }}</v-icon>  
2
```

プロパティスタイルガイド

- プロパティが多くなりやすいので長くなるときは改行して書く。その際v-textを使いMustacheをなくすと見やすくなる

```
1 <v-icon  
2   :disabled="String ? true : false"  
3   v-text="String ? 'mdi-star' : 'mdi-star-outline'"  
4 ></v-icon>  
5 |
```


スロット対応コンポーネント

- v-app-barはv-imgを含めることができ、その場合テンプレートスロットを経由してv-imgのプロパティをカスタマイズできる
- v-dialogはactivatorスロットが用意されているため、呼び出すボタンと呼び出されたダイアログを一つにすることが可能

イベント、関数について

- イベントが起こった際、コンポーネント内部にデータが保存されるため、名前を指定すれば取得、上書きすることができる
- valueプロパティや\$event引数を使用すればコンポーネント間でデータの受け渡しができるためデータオプション(methodsなど)を書かなくてよくなる

コンポーネントAPIについて

- とにかくドキュメントを読む
- 公式でAPI検索エンジンがある
- [//vuetifyjs.com/en/components/api-explorer](https://vuetifyjs.com/en/components/api-explorer)
- 名前が分かりやすいので英語でも読みやすい
- 出てきたAPIにも検索フォームがあるので便利

APIドキュメントについて

- 右上にコンポーネントソースとMD2に関するリンク
- ソースは見るだけではなく、Codepenリンクで動作チェックできるため試してみる
- 最近ではPlaygroundを用意しているAPIドキュメントも増えてきているので触ってみる
- 右の目次を見るとPlaygroundあるか分かる

主要コンポーネント1

- v-app、v-content
- [//vuetifyjs.com/en/components/application](https://vuetifyjs.com/en/components/application)
- v-contentでまとめたコンポーネントはappプロパティや一部コンポーネントの影響を受け良い感じに表示してくれるのでなるべく使用する
- v-appは必須

主要コンポーネント2

- v-container、v-row、v-col、v-spacer
- [//vuetifyjs.com/en/components/grids](https://vuetifyjs.com/en/components/grids)
- v-row、v-colはv1で分かりづらかったv-layout、v-flexから名前変更したもの
- Flexboxなのでレイアウト慣れている人は楽

主要コンポーネント3

- v-card、v-card-title、v-card-text、v-card-actions
- vuetifyjs.com/en/components/cards
- flatプロパティを使えばBootstrapに近いデザインになるのでマテリアルデザイン苦手な人は使う
- classを多用するのでdiv、p、spanを入れてもよい

Vuetify 2.0

- MD2対応に伴い、非常に多くの破壊的変更が入った
- アップグレードガイドが用意されていたり、consoleで分かりやすいエラーが表示されるので詰まることは少ない
- しかし変更点は本当に多い

new Vuetify

- 公式ドキュメントを見ると見慣れない記法を行っているが、Vue Routerと同じようにインスタンスを作成している
- なのでCDNの場合new Vueやnew VueRouterのように変数宣言してしまったほうが良い
- routerと同じくValueを省略して書くとなお良い

Vuetify Icons

- Google公式アイコンより materialdesignicons.min.css を推奨しているのでこちらに移行
- スネークケースからケバブケースに変更し、接頭辞 mdi- をつける必要がある
- また、アイコン名が若干違う

Material Design Icons

- ロゴアイコンが大量にあるのでこれ一つで完結
- 公式アイコン検索エンジンが重いので注意
- [//materialdesignicons.com/contributor/Google](https://materialdesignicons.com/contributor/Google)
- 先にGoogle公式アイコンを検索して、その後非公式アイコンを検索するとスムーズに移行できる
- materialdesignicons.com

Font Awesome

- v1で使用できたGoogle公式アイコン、Font Awesomeも引き続き使用可能
- new Vuetifyインスタンスでカスタムインストールも可能

TypeScript

- v2はJavaScriptからTypeScriptへの移行
- Vuetify自体の開発が楽になっただけで、Vue.jsやNuxt.jsのTypeScript対応と相性がいいわけではない
- @nuxt/vuetifyは対応したものの、使うとコケる人が多い

Sass + SCSS

- v1のStylusから移行
- Vue.jsのクラスとスタイルのバインディングと相性がいい(らしい)
- SCSSは_variables.scssで使用しており、定数の変更がv1より簡単になった

next Vuetify

- 今後はaccessibility (a11y) を軸にしたアップデートを行う
- 個人的には早くLighthouseで満点出したい。
- [//vuetifyjs.com/en/getting-started/roadmap](https://vuetifyjs.com/en/getting-started/roadmap)
- next.vuetifyjs.com

Expert Tips 1

- 背景色はv-applicationとv-application--wrapを変更する
- v2からnew Vuetifyインスタンスで変更
- JavaScriptの仕様上KeyをクォーテーションでくくってValueにカラーコード指定
- ちなみにv1は#appクラスのbackground使用可

Expert Tips 2

- v-modelを使う処理はスロットの調整がうまくいっていないため重い
- valueプロパティや\$event引数を使用したデータオプション(dataなど)を使わない処理にすれば軽くなる
- Vue.jsにあわせて将来的には改善される予定

Expert Tips 3

- v-btnとv-tabsはマテリアルデザイン準拠なのでアルファベットすべて大文字
- CSS3のtext-transformプロパティで処理しているのでクラスとスタイルを上書きできる
- 「V-SAPPORO」になる時「text-transform: none」スタイル指定すると「v-sapporo」に戻る

Expert Tips 4

- IE11とSafari9のサポートは頑張れば可能
- babel-polyfillを使えばできるものの、注意点あり
- 詳しくはQuick Startにて
- CDNでもbabel-standaloneで可能

Expert Tips 5

- `//cdn.jsdelivr.net/npm/@babel/standalone@7.5.5/babel.min.js`
- または
- `//cdn.jsdelivr.net/npm/babel-standalone@6.26.0/babel.min.js`
- あとは`type="text/babel"`つけて完了

Expert Tips 6

- Vue Loaderのようなwebpackローダーとして Vuetify Loaderが公式である
- Vuetifyの依存関係が複雑なので使うとコケることが多い
- ソースコード見ながら動的コンポーネントによる手動インポートしたほうが確実

Expert Tips 7

- ルーティングはVue.js公式のものを使う
- router.vuejs.org
- ユニットテストはVue.js公式のものを使う
- vue-test-utils.vuejs.org

Expert Tips 8

- ソースコードは複雑ではあるが非常に洗練されているので、コンポーネントを作成するうえで参考になる
- ドキュメント自体Vuetifyで作成されているため、コンポーネントを使用した開発の参考になる
- [//github.com/vuetifyjs/vuetify](https://github.com/vuetifyjs/vuetify)

Expert Tips 9

- Discordがある(i18n)
- community.vuetifyjs.com
- Meetupもある(東京)
- vuetify-meetup.connpass.com
- 参加するだけで楽しいVuetifyの世界へようこそ！

エキスパートVueify

ご清聴ありがとうございました！

追記

- connpass掲載にあたり、少し加筆修正しました
- Discordは「英語のみ」と紹介しましたが、日本語チャンネルが2019/08/02に開設されました
- このスライドはMIT Licenseです。ですが名称と画像の一部につきましては各権利者に帰属します。シェア大歓迎です。よろしくお願ひします