Firebase Vue.js で作る認証付き アプリケーション ハンズオン

環境

- Firebaseに登録しておいてください
- Vue.jsの開発にはこちらを使ってください
- GitHubとTwitterの認証はAPIkeyが必要です.

(場合によってはGitHubとTwitterは見せるだけとなることがあります)

ID: Naporitan

所属:情報系の学部4年生

趣味: ● プログラミング

- Vtuberを見ること
- 甘いもの

ID: Naporitan

所属:情報系の学部4年生

趣味: ● プログラミング

● Vtuberを見ること

• 甘いもの



(C)2019 CHARAT

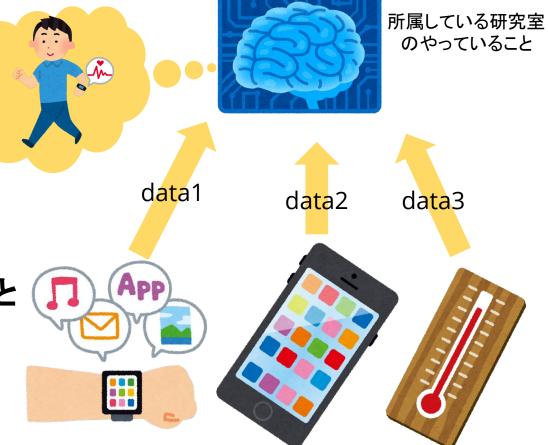
ID: Naporitan

所属:情報系の学部4年生

趣味: ● プログラミング

● Vtuberを見ること

• 甘いもの



ID: Naporitan

所属:情報系の学部4年生

趣味: • プログラミング

- Vtuberを見ること
- 甘いものPython: バイトの業務を自動化
 - JavaScript: この世のCallbackをPromiseに変えたい!
 - Vue.js: 趣味のWebApp作成
 - React: バイトが1週間前決まりました!

動的言語が好き!

ID: Naporitan

所属:情報系の学部4年生

趣味: ● プログラミング

• Vtuberを見ること

• 甘いもの



ID: Naporitan

所属:情報系の学部4年生

趣味: ● プログラミング

● Vtuberを見ること

• 甘いもの

タピオカジュースが めちゃめちゃデカかった!



Firebase * Vue.js

• なにがつくれるの?

フレームワークってなに?

● Firebaseで認証システムを作る利点は?

Firebase * Vue.js

• なにがつくれるの?

フレームワークってなに?

● Firebaseで認証システムを作る利点は?

なにがつくれるの?

Apple Store, Google Playのアプリと何が違うのか?

みなさん、わかりますか?

Webアプリ、ネイティブアプリの比較

Webアプリ

- DL必要なし!
- URLから集客が見込める!
- プラットフォームに支払い必要なし
- リリースに審査が必要ない!

ネイティブアプリ

- スマホの機能を最大限活かせる
- アプリ内課金が可能
- 動作が速い
- Push通知ができる

Webアプリ、ネイティブアプリの比較

Webアプリ

- DL必要なし!
- URLから集客が見込める!
- プラットフォームに支払い必要なし
- リリースに審査が必要ない!

ネイティブアプリ

- スマホの機能を最大限活かせる
- アプリ内課金が可能
- 動作が速い
- Push通知ができる

結局なにがつくれるんだよ!

思いついて 技術があれば なんでも

Firebase × Vue.js

• なにがつくれるの?

• フレームワークってなに?

● Firebaseで認証システムを作る利点は?

フレームワーク

出典: フリー百科事典『ウィキペディア (Wikipedia)』

フレームワーク

framework 「枠組み」「骨組み」「構造」などと和訳できる英単語。

- <mark>ビジネスフレームワーク</mark> 経営戦略や<mark>業務改善</mark>、問題解決などに役立つ分析ツールや思考の枠組み。MBAなどで教わることが多く、ビジネスに必要とされるロジカルシンキングや発想法などを体系的にまとめたもの。SWOT分析、ファイブフォース分析などが挙げられる。
- 開発・運用・意思決定を行う際に、その基礎となる規則・構造・アイデア・思想などの集合のこと。日本語では「枠組み」などと訳されることが多い。
- ソフトウェアフレームワーク コンピュータ・プログラミング用語。アプリケーションフレームワークなど。

flamework

バーナーワークの別称。

Wiki

フレームワーク

出典: フリー百科事典『ウィキペディア(Wikipedia)』

フレームワーク

framework 「枠組み」「骨組み」「構造」などと和訳できる英単語。

- <mark>ビジネスフレームワーク</mark> 経営戦略や<mark>業務改善</mark>、問題解決などに役立つ分析ツールや思考の枠組み。MBAなどで教わることが多く、ビジネスに必要とされるロジカルシンキングや発想法などを体系的にまとめたもの。SWOT分析、ファイブフォース分析などが挙げられる。
- 開発・運用・意思決定を行う際に、その基礎となる規則・構造・アイデア・思想などの集合のこと。日本語では「枠組み」などと訳されることが多い。
- ソフトウェアフレームワーク コンピュータ・プログラミング用語。アプリケーションフレームワークなど。

flamework

• バーナーワークの別称。

「完全に理解しているすごい人たち」 が 「効率のいい開発のノウハウ」 をコードレベルでまとめてくれたもの

家の鍵を想像してください 鍵の仕組みって知っていますか?

知らなくても、鍵は開けられますよね?

それが、フレームワークです

実際のコードを見てみましょう

今までのやり方

```
<!DOCTYPE html>
    <html>
 3
    <head>
    </head>
 6
    <body>
      <button id='btn'>Click</button>
 8
 9
      10
    </body>
11
12
    </html>
13
14
    <script>
15
      var btn = document.getElementById('btn')
      console.log(btn)
16
17
      btn.addEventListener('click', function () {
        document.getElementById('text').innerHTML = 'clicked'
18
      })
19
20
    </script>
```

Vue.jsなら

```
<!DOCTYPE html>
     <html>
     <head>
      <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
     </head>
     <body>
      <div id="app">
10
       <button @click="click">Click</button>
       {{msg}}
11
12
      </div>
     </body>
14
     </html>
16
    <script>
17
      new Vue({
18
19
        el: '#app',
20
        data () {
21
          return {
22
            msg: ''
23
24
25
        methods: {
26
          click(){
27
            this.msg = 'clicked'
28
29
      })
30
31 </script>
```

フレームワークすごい

Vue.jsの方が、長いコードになっている しかし

圧倒的に読みやすく、わかりやすい

Firebase * Vue.js

• なにがつくれるの?

フレームワークってなに?

Firebaseで認証システムを作る利点は?

Firebaseを使わない時はどうやるの?

軽く説明します

ユーザー登録時

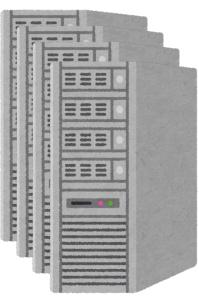
暗号化アルゴリズム

プレーンテキスト

userName passWord



Webサイト



WebServer

暗号化済

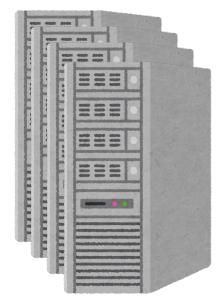
userName passWord



DataBase

ユーザー登録時

暗号化アルゴリズム



WebServer

password(プレーンテキスト)



暗号化アルゴリズム でハッシュ化



ソルト



ハッシュ化 されたPW

Passwordの保存のされ方



DataBase

User	password	solt
User1	\$2a\$10\$2Z2DslKcjhc.4ry zhu4aoObT96oQVX1gu. QqlyTTnTwk691Ln0.NC	\$2a\$10\$2Z2DslKcjhc.4ry zhu4aoO
User2	\$2a\$10\$CBel.QyhtwnN8 APZsNi86uxjjZMPb8GVI m0ANwqbP7pyq.9rZXhl.	\$2a\$10\$CBel.QyhtwnN8 APZsNi86u

Passwordの保存のされ方



DataBase

User	password	solt
User1	\$2a\$10\$2Z2DslKcjhc.4ry zhu4aoObT96oQVX1gu. QqlyTTnTwk691Ln0.NC	\$2a\$10\$2Z2DslKcjhc.4ry zhu4aoO
User2	\$2a\$10\$CBel.QyhtwnN8 APZsNi86uxjjZMPb8GVI m0ANwqbP7pyq.9rZXhl.	\$2a\$10\$CBel.QyhtwnN8 APZsNi86u

password(プレーンテキスト)

Passwordの保存のされ方



DataBase

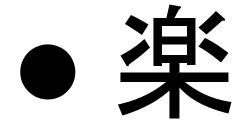
User	password	solt
User1	\$2a\$10\$2Z2DslKcjhc.4ry zhu4aoObT96oQVX1gu. QqlyTTnTwk691Ln0.NC	\$2a\$10\$2Z2DslKcjhc.4ry zhu4aoO
User2	\$2a\$10\$CBel.QyhtwnN8 APZsNi86uxjjZMPb8GVI m0ANwqbP7pyq.9rZXhl.	\$2a\$10\$CBel.QyhtwnN8 APZsNi86u

firebase(プレーンテキスト)

Firebaseで作る利点

● 楽

Firebaseで作る利点



Firebaseで作る利点

- 暗号化技術を熟知してなくてOK
- ユーザ認証のDBの設計に悩む必要なし
- SDKが揃っている
 - 早く、簡単に実装できる
 - 様々で言語でアプローチができる

使わない理由なくないですか?

そのための

「今日」

ですよね?

やっていきましょう!

休憩[5-10分(らい)

後半

- Vue.jsについて
- コンポーネントを作ってみる
- 実際にloginシステムを使ってみる

Vue.js

- html, JS, cssが一体化したファイルを作っていく
 - コンポーネント単位で開発していく
 - 一体化したファイルをコンパイルして使う
- htmlにJSをかける
 - htmlでfor, ifが使えたり・・・・
 - eventを簡単に設定できたり・・・・
 - 変数をレンダリングできたり・・・・
 - htmlとJSの双方向データ受け渡しが簡単

Vue.js

- html, JS, cssが一体化したファイルを作っていく
 - コンポーネント単位で開発していく
 - 一体化したファイルをコンパイルして使う
- htmlにJSをかける
 - htmlでfor, ifが使えたり・・・・
 - eventを簡単に設定できたり・・・・
 - o 変数をレンダリングできたり・・・・
 - htmlとJSの双方向データ受け渡しが簡単

Vue.jsのライフサイクル

Vue.jsのライフサイクル

一体化しているってなんだ?

E-mail: E-mail naporin0624@gmail.com これをこう! **PassWord** Password: login By Email login By Twitter submit login By GitHub MyForm.vueという 単一ファイルで このFormを実現できる

一体化しているってなんだ?

```
<template>
                                                               <script>
                 <div class="my-form">
                   <label>E-mail:</label>
                   <input type="text" v-model="email">
                   <label>Password:</label>
                   <input type="password" v-model="password">
                   <button @click="submit">submit</button>
                </template>
                E-mail:
naporin0624@gmail.com
              Password:
```

submit

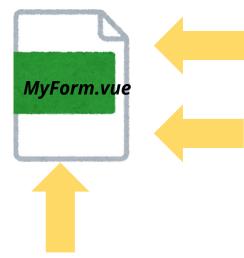
```
export default {
                                        display: inline-grid;
  name: 'MyForm',
  data () {
     return {
       email: '',
       password: ''
                                        border: none;
  methods: {
                                        background: none;
                                        background-color: ■gainsboro;
    submit () {
       output = 'email: '
                                       .my-form button {
       output += this.email
       output += '\n'
       output += 'password: '
                                        background: none;
       output += this.password
                                        background-color: □lightgoldenrodyellow;
       console.log(output)
                                       .my-form button:hover {
                                        background-color: ■grey;
                                       .my-form button:active {
                                        background-color: ■yellow;
</script>
```

.my-form {

一体化しているってなんだ?

<script>

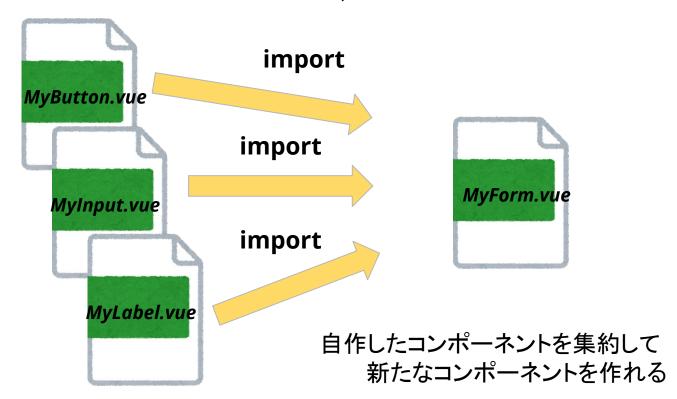
全てMyForm.vue に書かれている



```
email: '',
     password: ''
 methods: {
   submit () {
     output = 'email: '
     output += this.email
     output += '\n'
     output += 'password: '
     console.log(output)
</script>
```

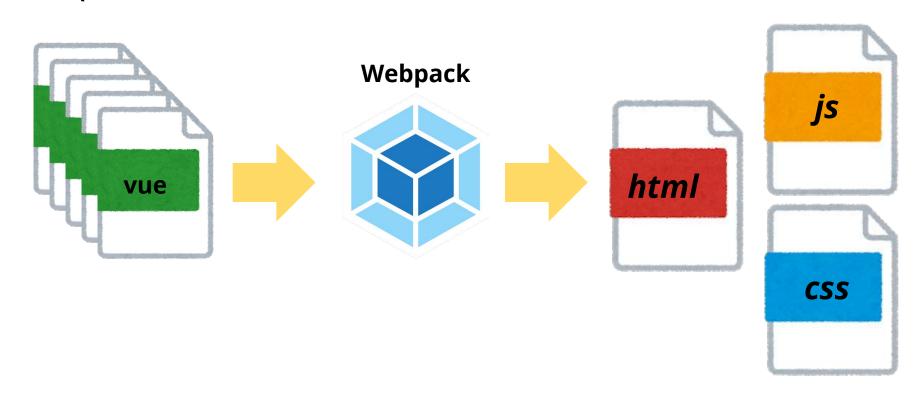
```
my-form {
display: inline-grid;
mv-form input {
 border: none;
 background: none;
 background-color: ■gainsboro;
my-form button {
 border: none;
 background: none;
 my-form button:hover {
 background-color: ■grey;
.my-form button:active {
 background-color: ■yellow;
```

一体化しているから、できること

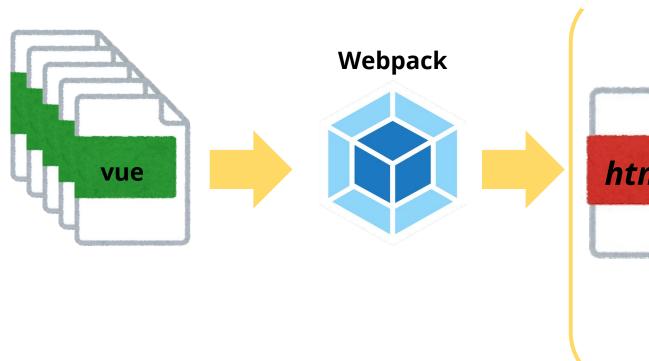


weファイル、どうやって動かす?

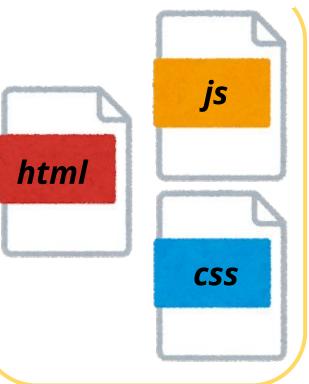
Webpackによるコンパイル



Webpackによるコンパイル



これを本番環境で動かす



Vue.js

- html, JS, cssが一体化したファイルを作っていく
 - コンポーネント単位で開発していく
 - 一体化したファイルをコンパイルして使う
- htmlにJSをかける
 - htmlでforが使える・・・・
 - eventを簡単に設定できたり・・・・
 - 変数をレンダリングできたり・・・・
 - htmlとJSの双方向データ受け渡しが簡単

html/こJSが・・・! ?

```
押してね
押してね
押してね
押してね
押してね
押してね
押してね
```

コンパイル

```
<template>
 <div class="hello">
   <div class="button-loop">
      <my-button
        v-for="n in 10"
        :key="n"
       @click="func"
       押してね
      </my-button>
    </div>
  </div>
  template>
```

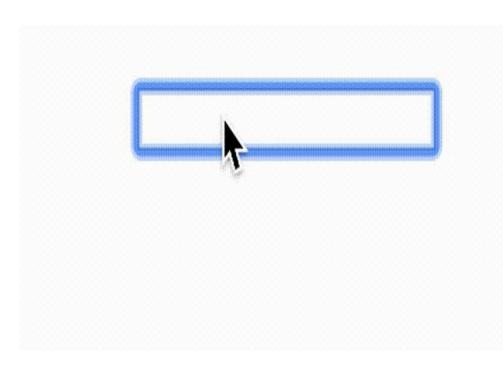
htmlにJSが・・・! ?

Button1 Button2 Button3 Button4 Button5 Button6 Button7

コンパイル

```
<template>
 <div class="hello">
   <div class="button-loop">
     <my-button
       v-for="n in 7"
        :key="n"
       @click="func"
       button{{n}}
     /my-button
   </div>
 </div>
```

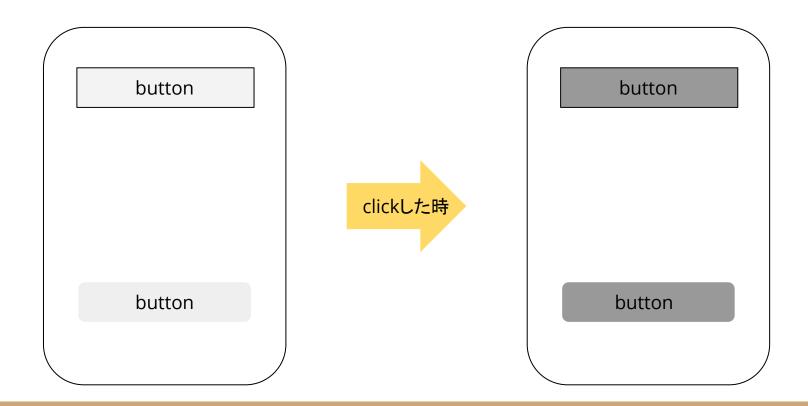
双方向バインディング・変数レンダリング



```
<div class="hello">
   <input type="text" v-model="msg">
   {{msg}}
 </div>
<script>
export default {
 name: 'HelloWorld', You, 20 hou
 data () {
   return {
     msg: ''
</script>
```

Vue.jsの説明 おわり

Button.vueを作ってみる

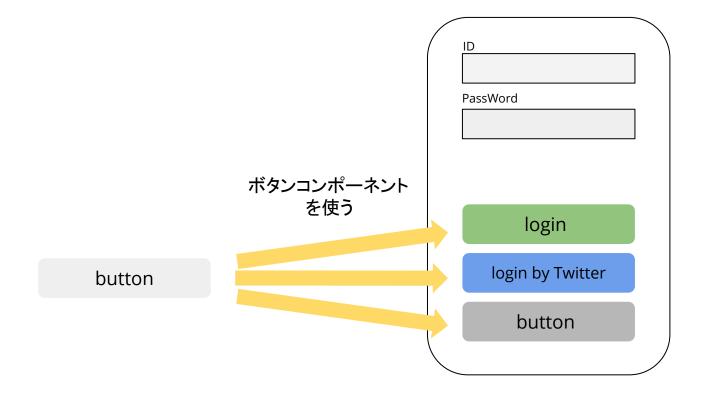


実際に作ってみましょう

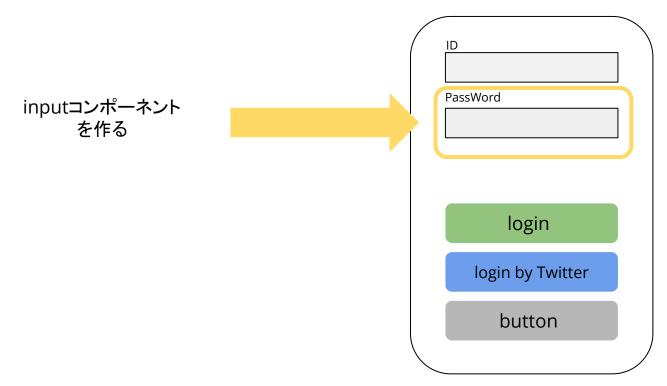
これをsandboxで動かします

Vueプロジェクトのgitリポジトリ

Form.vueを作ってみる



Form.vueを作ってみる



ログイン部分を実装する

ここは実際に一緒にプログラミングしていきましょう

いかがでしたか?

これで今回のハンズオンは終了です

よろしければ、アンケートにご協力ください!

アンケートはこちらから!